REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
19-12-2005	Final Report	25 April 2005 - 25-Oct-05

4. TITLE AND SUBTITLE

Robust Networking with Mobility

5a. CONTRACT NUMBER
FA8655-05-M4026

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dr. Kevin J Baughan

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Ideas Network Ltd / Birmingham University
42 Horseguards Drive
Maidenhead
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
SPC 05-4026

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Research will test the hypothesis that the R3 routing architecture can be used to provide robust networking with reasonable levels of mobility. The hypothesis will be tested by using a network simulation tool (OPNET™) to measure the performance of a network of static and mobile nodes and emulated wireless links. The goal is not to simulate the wireless environment but to model the impact of mobility on routing over such an environment. The simulations will cover:

- Intermittent links
- Links of widely varying capacities
- Changing topologies as a result of reasonable mobility – in our context this is defined as 'localised' mobility as far as the logical network abstraction is concerned, which will be mapped to real node mobility in a wireless environment
- Controlled changeovers from old to new topologies, and
- The impact of inconsistent network information

Performance comparisons based on measures such as throughput, delay and packet loss will be made between R3 and IS-IS in dealing with reasonable levels of mobility. The goal will be to show that R3 can provide a significantly more robust mobile networking infrastructure than is possible with existing SPF based algorithms.

15. SUBJECT TERMS
EOARD, C3I, Ad-hoc Networks, Wireless Networks

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			PAUL LOSIEWICZ, Ph. D.
UNCLAS	UNCLAS	UNCLAS	UL	60	**19b. TELEPHONE NUMBER** *(Include area code)* +44 20 7514 4474

Resilient Recursive Routing (R^3)

Towards Robust High Performance Networking

Final Report

FA8655-05-M-4026 & FA8655-04-M-4037

Summary

The purpose of this report is to present the combined results of two research activities sponsored by EOARD which have evaluated the potential for significantly enhancing **network robustness** based on a new routing technology R^3 (Resilient Recursive Routing) that is the result of a collaboration between Ideas Network Ltd and the University of Birmingham. The two research activities have looked at how robust networking can be achieved through:

- Improved network design, by providing a demonstration of how intuitive tools based on R^3 can enable operations staff in a command centre to deploy and maintain highly resilient networks
- Improved protocol design, by simulating how R^3 provides a powerful and scalable solution for mapping fixed and mobile traffic requirements on to a rapidly changing wired and wireless network infrastructure

In order to collect the necessary information, the research was carried out by using a network simulation tool (OPNET™ provided under the US Air Force Licence) and a mathematic modelling and analysis tool (MATLAB™). The simulation results have established that R^3 is able to demonstrate significant improvements in performance and resiliency over conventional SPF (Shortest Path First) based proactive/reactive routing algorithms (e.g. ISIS, AODV) in the following situations:

- A core network with various level of traffic loading
- A scale-free network suffering from symmetric or asymmetric attacks
- A wireless network with low degrees of node mobility suffering from attrition and communication link instability.

The team supporting these two research activities was composed of:

- The University of Birmingham
    - Dr Costas Constantinou
    - Dr Theo Arvanitis
    - Dr Sanya Stepanenko
- Ideas Network Ltd
    - Prof Kevin Baughan
    - Dr Bin Liu
    - Yu Sun (summer intern)

1. LNA -- the philosophy of R^3 neighbourhoods

1.1. Diversity and resilience measures on complex networks

In its Report to Congress in July 2001, the DoD (Department of Defence) described Network Centric Warfare (NCW) as "no less than the embodiment of an Information Age transformation of the DoD." The report went on to identify the fundamental tenets of NCW as follows:

- A robustly networked force improves information sharing;
- Information sharing enhances the quality of information and shared situational awareness;
- Shared situational awareness enables collaboration and self-synchronization, and enhances sustainability and speed of command;
- These, in turn, dramatically increase mission effectiveness.

A considerable amount of research has gone into many aspects of NCW, but very little has gone into really understanding the principles of **robust networking** – the fundamental premise upon which NCW is built. Without an underlying infrastructure that can deliver robust networking, none of the other benefits of NCW are realisable. Indeed, a military force that is constructed around NCW could be at a significant military disadvantage if their networking capability was seriously disrupted or compromised.

In order to achieve robust networking, a fundamental question is how to measure diversity and resilience in complex networks, where topologies are highly irregular. Actually, the analysis of connection diversity and clustering in large-scale complex networks is important for many applications in areas of science spanning social science, biology, physics and engineering. Quantifying such notions is challenging and no general consensus exists, as external and often arbitrary criteria are applied in clustering techniques.

LNA (Logical Network Abridgement) provides a fundamentally new approach and introduces a method of quantifying the diversity of paths in a network without resorting to clustering or the imposition of arbitrary additional criteria. LNA is an iterative abstraction technique which is based on the notion of a cycle as the elementary unit of diversity that enables the natural adoption of diversity and clustering metrics (i.e. the definition of neighbourhood) arising from the topology of the network alone. This work differs from previous studies in that it quantifies notions of diversity and consequent resilience, not just in terms of the physical connectivity characteristics of an end-to-end path (e.g. the diameter of a network[1]), but in terms of the richness of end-to-end path diversity (i.e. all distinct paths available for a given network topology). Based on this

[1] R. Albert, H. Jeong, A.-L. Barabasi, *Nature* **406**, 378 (2000)

method, the path efficiency, connection diversity and network resilience can be measured and predicted at complex networks.

1.2. LNA review

When considering the connection diversity and thus resiliency in a network, it is important to quantify the number of distinct paths between any pair of end-nodes. The loss of one path is insignificant if numerous other paths exist. At the other extreme, if only a single path exists, loss of any of its component nodes or links results in the network becoming disconnected into two independent sub-networks. The simplest and most elementary form of diversity is when two disjoint paths connect two nodes, i.e. these nodes belong to a ring, or cycle in graph-theoretic terminology (see Fig. 1.1A). We shall refer to such a topological relation, as a simple neighbourhood and all nodes belonging to the same cycle are thus neighbours. Any graph with n nodes and e edges has $v = e - n + 1$ independent cycles (v is the cyclomatic number of a connected graph[2]). The concept of independence in cycle space is defined relative to the binary addition of cycles. The binary addition (or symmetric difference) of two cycles is the set of edges, which are in either cycle, but not in both. This operation is the set-theoretic equivalent of the XOR operation in Boolean logic. The choice of a basis set of cycles is not unique as we shall see shortly.

Every independent cycle or neighbourhood of nodes can be abstracted to a logical node (e.g. the grey node in Fig. 1.1A), intended to represent a diversity unit. Two cycles are defined to be adjacent (in a diversity sense) if they share at least one common edge (e.g. the edge and its incident nodes highlighted in dotted black in Fig. 1.1B). The nodes incident to the common edge are gateway nodes between the two cycles. Connecting logical nodes (e.g. the grey nodes in Fig. 1.1B) with their associated logical edges (e.g. the grey edge in Fig. 1.1B), we can construct the next logical level abstraction of the network. If the abstracted logical level description of the network contains cycles, we can repeat the above procedure as many times as required, or until it terminates in a highest-level loop-free logical network structure. In Fig. 1.2 we have the original physical level (level $l = 0$), and logical levels $l = 1$ and $l = 2$ (the latter being trivially a single logical node rather than a tree). We call this procedure of recursive abstraction logical network abridgment (LNA). We label nodes as $l.n$ where l denotes the level of abstraction and n is the node number at that level. Thus, 1.2 is node 2 at level 1 (identified with the cycle 0.2-0.4-0.5-0.2 at level 0 in Fig. 1.2). It is worth pointing out that when we refer to the LNA abstraction we signify the entire ensemble of levels.

Since the choice of basis cycle set is not unique, it follows that the LNA procedure is also not unique, as it is dependent on this choice. Additional criteria suited to the problem or application at hand need to be employed to make the choice of basis cycle set deterministic. For the purposes of our discussion we choose to minimise the number of

[2] R.Diestel, *Graph Theory*, vol. **173** Graduate Texts in Mathematics (Springer, Berlin, 2000)

gateways (i.e. number of logical links at the next level[3]). For planar graphs the choice of independent cycles is simpler, as these can be identified with the faces of a particular embedding of the planar graph.

Every level of abstraction conveys summarised path diversity information for the previous level, which can aid both the visualisation and analysis of this diversity. The summarisation is not done on an arbitrary clustering basis, but is dictated by the underlying network topology and introduces a natural metric for the network, the levels of abstraction. Clearly the bigger the number of levels of abstraction until a loop-free highest level is derived, the more the intrinsic path diversity exists in a network. If the graph at any level of abstraction becomes disconnected, this indicates the existence of a path diversity bottleneck at the previous level. An example of the application of the LNA procedure to a graph illustrating the above point is shown in Fig. 1.3.

1.3. LNA to communication networks – R^3 (Resilient Recursive Routing)

In this report, we focus on the application of the LNA to communication networks, specifically to routing in packet-switched networks such as the Internet.

The network diversity index, D, is the simplest global measure of diversity in a network and can be defined as, $D \equiv L/N$; $0 \leq D \leq$ Lclique/N, where L = lmax is the number of levels of abstraction and N is the number of nodes in the network. We observe that L is strictly bounded between 0 for a tree and Lclique. Lclique may or may not be a finite number, but this does not affect the discussion or conclusions that follow. In the context of a communication network, the diversity index D can be used to determine the type of routing protocol best suited to the network. If $D \rightarrow 0$, the network is dominated by trees and a shortest path type protocol is highly scalable and efficient. At the other extreme, if $D \rightarrow$ Lclique/N, the network is very close to fully meshed and random deflection routing is scalable, robust and sufficient, because if a destination is not reachable directly there is a high probability that it can be reached through any one of the neighbouring nodes. Away from these two extreme cases a shortest path type protocol fails to exploit the underlying network diversity and will take time to re-converge if congestion or failures arise, while on the other hand random deflection routing is unlikely to result in the successful delivery of data to its intended destination, as nodes are likely to be separated by many hops. To exploit the underlying network diversity a dynamic, adaptive routing protocol is then required.

We illustrate how the LNA can be exploited to create an adaptive resilient routing protocol for a communication network with an intermediate value diversity index D. A

[3] F. Berger, *Minimum Cycle Bases In Graphs* (Shaker Verlag, Aachen, 2004)

fundamental routing algorithm can be constructed such that it is capable of loop-free routing a data packet around a cycle of physical nodes, by attaching a label to a packet that defines explicitly or implicitly an arc on the cycle. The algorithm can also send probe packets to measure the performance or levels of congestion of each hop around the cycle in each direction of circulation. The probe packets can also be used to inform all nodes in the cycle of any failures in nodes or links, thus always ensuring a fast reroute reaction. For source and destination nodes belonging to the same physical cycle, this algorithm is invoked once. For source and destination nodes belonging to different cycles, and lying on the same cycle at logical level 1 the algorithm needs to be invoked twice, once at each level. For source and destination nodes belonging to the same cycle at level *l*, the algorithm needs to be invoked $l + 1$ times. The recursive nature of this algorithm ensures that all properties such as load balancing and fast reaction to failures, which were implemented at the physical level (i.e. are local properties), scale automatically globally. The only caveat is that the dissemination of summarised measurement performance information to higher logical levels is done on slower frequency to bound strictly the measurement overheads. However, this slower frequency is consistent with the much slower periods over which networks of increasing size are manually managed. We call such a routing scheme *resilient recursive routing (*R^3*)* and present its concepts and implementation at the following sections of this report.

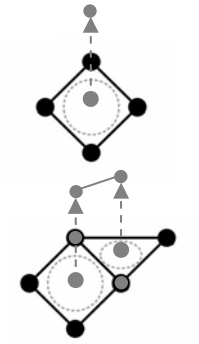


Fig.1.1 (A) Simplest form of path diversity in a graph is a simple cycle, which is abstracted to a logical node.

(B) Definition of a connection (adjacency) between two logical nodes.

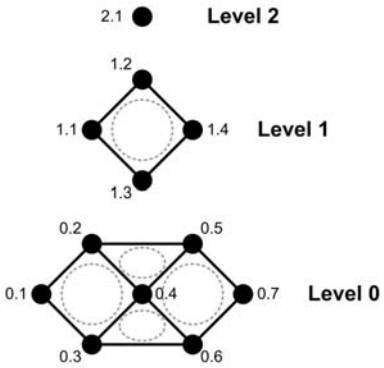


Fig. 1.2 The logical network abridgement procedure (LNA) applied to a simple network. The LNA abstraction is the ensemble of levels 0, 1 and 2. Physical cycles at level 0 are identified as logical nodes at level 1; common links between cycles at level 0 correspond to logical links at level 1; and the abstraction is iterated until a highest level 2 (loop-free) graph is arrived at. The labelling of nodes is in two parts, the first one corresponding to the level of abstraction and the second enumerating the node at this level.

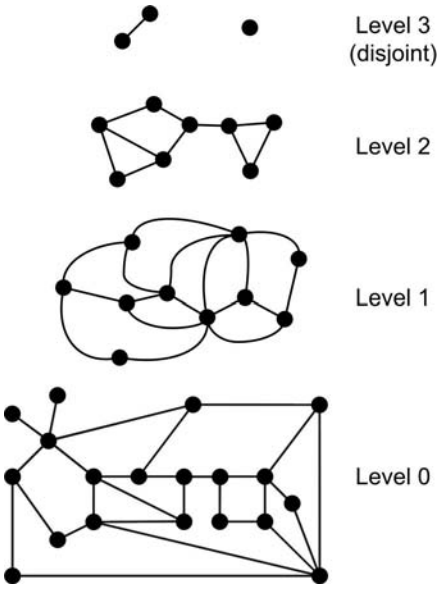


Fig. 1.3 The logical network abridgment abstraction of a graph which results in a disjointed logical level 3. The disjoint nature of logical level 3 is a characteristic signature of reduced path diversity between more highly connected clusters in the physical level 0 network (i.e. path diversity is not homogeneous across the network) and must not be confused with the absence of connectivity.

2. LNA as a network visualization and design tool

In the first research activity a demonstrator visualization tool was developed in order to illustrate in a visual manner the R^3 abstraction framework for particular real-world example networks. The purpose of the visualization tool being to allow operational staff the facilities to assess the 'health' of the network in terms of robustness and resiliency in case of any attrition that might have occurred. Furthermore, the demonstrator showed the potential of this tool to become a decision-support facility for designing or modifying networks for any communications infrastructure scenario including Defense Command & Control (C2). The main features and functions of the tool include:

1. The provision of 2D and 3D views, from physical level to high abstract level, for the topology of the network;
2. The representation of the status of the network, in term of:
    a. Showing on request the vulnerability of the network based on the information relating to the R^3 the abstraction;
    b. Showing on request the network capacity from the highest abstract level to the physical level;
    c. Providing alternative solutions to adjust the topology or functionality of the network, when the network is attached or congested.
3. The customization flexibility of the tool for changing the topology of network by removing/adding nodes/links, or changing the status and type of a node and/or a link.

As discussed before, the R^3 abstraction algorithm works independently of any specific communication protocols used in a network and enables the depth of the resiliency of the network to be identified. The visualization demonstrator enables this by providing 3-dimensional views of the levels of abstraction, allowing C2 operators to access in a glance the depth of resiliency. Fig. 2.1, shows an example of a network, abstracted with the R^3 framework and visualized in 3D. The number of levels in the network represents the networks depth of resiliency.

The visualization tool allows users to toggle between 3D and 2D views of the network, while they can examine each individual level of abstraction and the overall 2D or 3D structures with a variety of visualization functions, such as zoom and rotation. The use of appropriate 3D graphical object representations for each element of the network (e.g. core router, edge router, optical/cable based link etc.) and appropriate colours for describing the status of these elements, allow networks operators to fully understand the 'health' of the network in terms of congestion, lost or attacked links, etc.

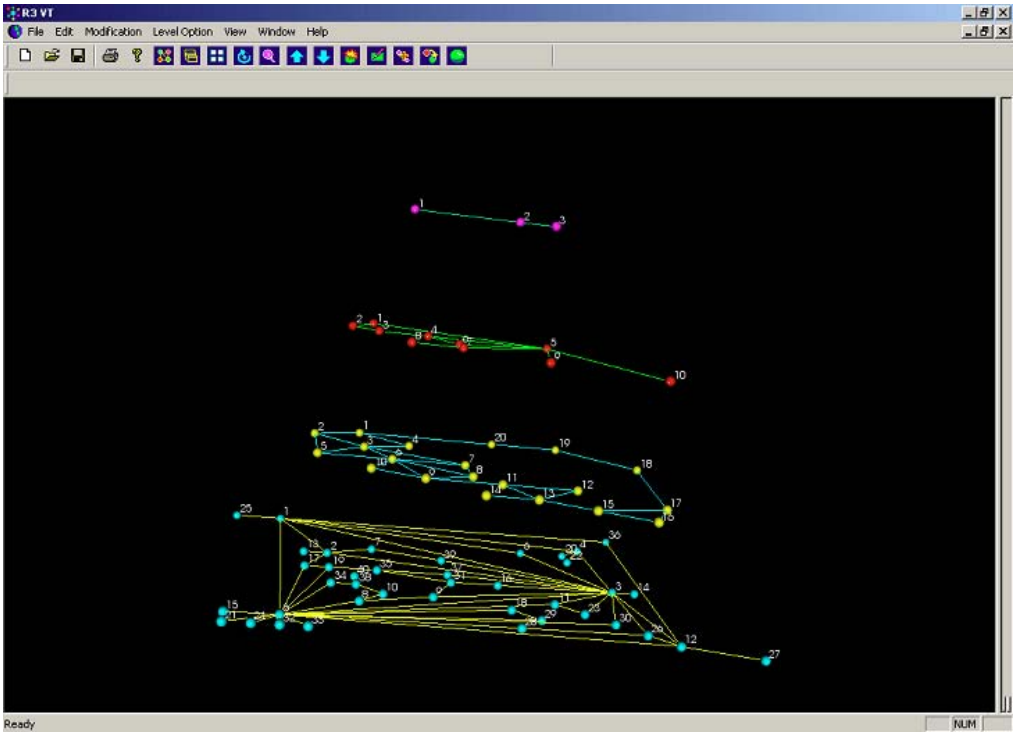


Fig. 2.1: *A 3D visualization of the* R^3 *abstraction algorithm in an example network. The lowest level of the abstraction will represent the physical layer of the network. The remaining 3 levels represent the recursive abstraction of the network. The final level represents a loop-free deterministic tree.*

Within the visualization tool it is possible to have a quantitative view of various parameters in the network (i.e. congestion level of links), and to calculate the network's vulnerability, which is defined by a simple index. The index provides a measure of the impact of loosing a specific node (or link) based both on how much the network 'shrinks' in terms of connectivity and how much it looses in terms of its overall depth of resiliency, (and hence how much it looses in terms of path diversity across the network). The following expression represents the vulnerability index used by the tool:

$$V_{index} = \frac{Number\_of\_Nodes\_before}{Number\_of\_Nodes\_after} * \frac{R3\_levels\_before}{R3\_levels\_after}$$

The highest the vulnerability index is for a node the most vulnerable the network will become in case this node is lost. Fig. 2.2 shows the vulnerability index of 5 nodes on the example network we have used above.

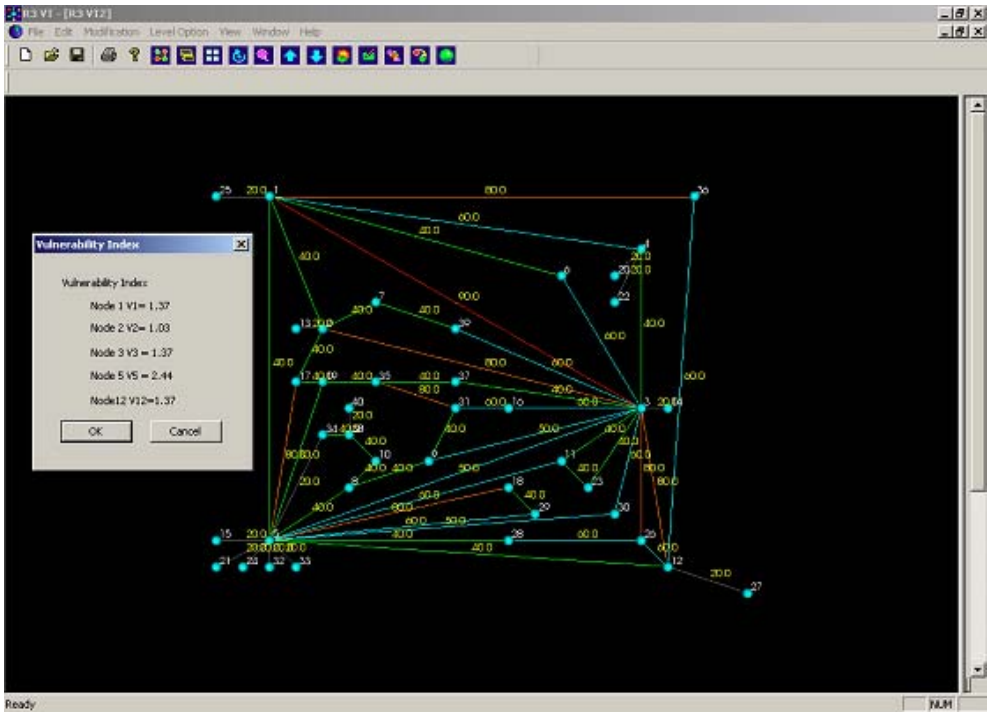


Fig. 2.2: *This figure shows the calculation of the vulnerability index for 5 nodes in the network and visualizes the congestion level for all links.*

Fig. 2.3: *An example of an attack and recovery scenario.*

Finally, the various customization facilities of the tool can allow the operator to re-design the network and recalculate its resiliency and vulnerability. In addition, for this demonstrator we illustrate how R^3 can automate the re-calculation of the levels in a network that has been attacked. Fig. 2.3 shows an example.

The above elements show that the R^3 technology allows C2 operators within a Network Centric Warfare (NCW) context to visually assess a C2 communications infrastructure, and then design and manage more effective new solutions.

The R^3 VT demonstrator is written in C++, and developed on Microsoft Visual Studio.NET 2003. R^3 VT uses the VTK (Visualization Tool Kit 4.2) library for the 3D visualization. A CD-ROM with the full source and compiled code, as well as a user and developer's manual was delivered separately to EOARD as part of the interim milestone of the first research contract on 17 September 2004.

3. Resilient Recursive Routing (R^3)

3.1. R^3 detailed review

3.1.1. Desirable protocol properties

Our philosophy in seeking a truly adaptive routing protocol adheres to elementary principles of distributed adaptive systems of autonomous nodes[4], complemented by key concepts that address a number of fundamental questions.

To begin with, we seek a connectionless routing solution that does *not attempt to micro-manage traffic flows*. Therefore, we admit traffic belonging to a number of priority classes and try to take advantage of the gains afforded by exploiting economies of scale due to statistical multiplexing as far as possible, without resorting to any resource reservations. We do this to provide a complementary solution to existing connection-oriented networking techniques, in order to provide a scalable solution that is capable of automatically optimizing performance of highly dynamic traffic flows. Such traffic flows could be associated with large numbers of broadband hosts on a public network, large numbers of enterprise hosts on a private network, or large numbers of military hosts on a defense network.

In order to maximize the efficient use of network resources, we require that the routing protocol *adapts dynamically over a range of time-scales* pertinent to the network topology, as well as the offered traffic. These time-scales range from being sufficiently short in order to respond appropriately to congestion or failure, to the opposite extreme of being long enough to characterize the long-term variation of aggregate traffic over the large-scale topology of the network and even changes in the network topology itself.

Adaptation needs to be implemented in such a way that the autonomous operation of switching nodes is *always based on information that is timely with regard to the adaptation timescale*. For example, we require that information driving path selection decisions in response to local congestion or failures originates from the local *neighborhood* of each node, in order to ensure that all such decisions are accurate and can be altered as fast as required. Information from increasingly distant nodes, or neighborhoods of nodes, is used in a summarized form to inform the adaptation processes running over increasingly longer time scales.[5]

Returning now to the issue of making maximum efficient use of network resources, we

[4] A.S. Tanenbaum and M. van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, Upper Saddle River, NJ, 2002

[5] It is worth pointing out that the similarities with the philosophy of the fish-eye state routing protocols in mobile *ad hoc* networks are only superficial, as these do not associate different time-scales with different levels of coarse-grained descriptions of the network.

question the wisdom of shortest path or lowest cost routing and their variants in the light of the above desirable protocol properties. The majority of popular existing routing protocols[6] attempt to build a global topological picture of the network as a graph at each individual node, which they then proceed to decimate into a shortest path tree to all reachable destinations, thus discarding diversity information required to implement any form of adaptation. Multipath routing protocols[7] try to improve on this approach by discovering and maintaining a limited list of (often) disjoint paths to the destination, which are also computed using global information. By necessity, computing such end-to-end paths relies on increasingly out-of-date information from ever-distant nodes, thus rendering the output of algorithms such as Dijkstra's[8] unsuitable for use in a truly adaptive manner, unless scalability with respect to the number of nodes in the network is sacrificed. An oft employed way around the scalability issues is to introduce clustering and thus a topological hierarchy into the network[9], at the expense of inadvertently introducing performance bottlenecks (e.g. area border gateway routers) at the same time.

We choose to dispense with the concept of global optimality altogether, and wish to consider locally optimal routing decisions made only in the context of neighborhoods. Furthermore, we reject the use of any algorithm that decimates the frequently rich path diversity of a typical network graph into a tree, to facilitate the operation of adaptive routing algorithms as far as is allowed by the underlying network topology itself. Therefore, we need to exploit the presence of rings (i.e. neighborhoods) in the topology and at the same time engineer candidate routing protocols so as to be inherently free of data traffic loops. All this points towards a requirement to have a deterministic abstraction of the network topology, so that any network topology can be reduced to a set of neighborhoods over which effective routing decisions can be made on appropriate time scales.

3.1.2. A generic R^3 routing algorithm

The logical network abridgement can be augmented with a number of forwarding rules to create the resilient recursive routing protocol. Here we consider the high-level generic features of such a protocol that adheres to the properties discussed in the previous section. There can be more than one specific implementation of the generic algorithm and we shall describe our specific choice, which we have proceeded to simulate in the next section.

---

[6] R. Callon, Use of OSI IS-IS routing in TCP/IP and dual environments, IETF, RFC 1195 (1990); G. Malkin, RIP version 2, IETF, RFC 2453 (1998); J. Moy, OSPF version 2, IETF, RFC 2178 (1997)

[7] E.L. Lawler, A procedure for computing the K best path solutions to discrete optimisation problems and its application to the shortest path problem, Management Science **18** (1972) 401 – 405; E. Oki, A disjoint path selection scheme with SRLG in GMPLS networks, Proc. of IEEE HPSR'2002 (2002) 88 – 92

[8] T. Cormen, C. Leiserson and R. Rivest, Introduction to Algorithms, MIT Press, Cambridge MA, 1990

[9] F. Amer, and Y-N. Lien, A survey of hierarchical routing algorithms and a new hierarchical hybrid adaptive routing algorithm for large scale computer communication networks, Proc. of IEEE Int. Conf. on Communication (ICC) **2** (1988) 999 – 1003

The routing algorithm must operate recursively at each level of the network, either to route a packet around a single ring, or along a tree. Routing information on a tree is a trivial exercise, in the sense that all forwarding decisions are deterministic and we shall not discuss this any further. Our fundamental algorithm routes a packet from a source to a destination, both of which are members of the same ring (hereafter referred to as level 1 neighbors), and must be capable of (i) loop-free data routing across the ring, (ii) load balancing across the ring and (iii) fast reaction to link or node failures in the ring.

If the source and destination are neighbors at level 2, i.e. they are both members of the same level 2 ring, we *iterate* the fundamental routing algorithm first at level 2, and then at level 1 for the selected level 2 path. This enables us to route a packet in a loop-free manner, while performing load balancing and enabling failure recovery across the network. The only difference is that the characteristic reaction time of the fundamental routing algorithm to congestion and failures at level 2 will be based on summarized information over a longer time-scale to reflect the summarized nature of this higher-level neighborhood and to ensure scalability.

For a level n destination, we similarly iterate the fundamental algorithm at levels n, n-1, …, 1 in order to ensure that all the routing protocol properties scale across the entire network. If on average a neighborhood contains k nodes a simple, worst-case counting argument can show that a level n node will correspond to at most k^n physical (level 1) nodes. If the adaptation/update time constants determining the operation of the fundamental routing protocol at level n 'slow down' exponentially (i.e. are of the form $\tau_n = \tau_0 \cdot b^n$ for some τ_0 and b), we can then guarantee the scalability of the protocol adaptation overheads with increasing network size. The longest time scale can be chosen to be of the order of hours, days, weeks, or even months, whereas the shortest time scale needs to be of the order of tens or hundred of milliseconds.

Naturally, the adaptation can be 'terminated' at an earlier level of abstraction and the higher-level iterations of the fundamental routing algorithm can become static, if the network operation is deemed to be sufficiently adaptive by the protocol designer. More details see Appendix C – A specific R^3 routing algorithm implementation.

3.2. R^3 protocol prototype development

A series of R^3 routing protocol versions have been developed in C/C++ code based on the R^3 routing algorithm explained above. The major differences between each version are highlighted at Table 3.1. By the end of these two EOARD projects, R^3 v9 can achieve the following functions:

- Using routing protocol independent methods to explore network topology.
- Upon a limited set of topology types, an R^3 LNA architecture can be generated dynamically from scratch.
- Upon a topological change at any LNA level, an R^3 LNA architecture can be updated automatically (but not regenerated from scratch).
- Given an R^3 LNA architecture, all available routes for each pair of source – destination in the network can be generated automatically.
- An optimal route can be chosen dynamically upon the updated information of network performance.
- Network performance (e.g. end-to-end delay) can be measured at both physical level and logical level 1 at different time-scales.
- A label switching forwarding engine can be applied, where a structure of labels is inserted to a user packet. Routing decision based on such a label structure is connection-oriented in terms of routes, but connectionless in terms of paths.

	R^3v1	**R^3v2**	**R^3v3**	**R^3v4**	**R^3v5**	**R^3v6**	**R^3v7**	**R^3v8**	**R^3v9**
R^3 routes initialisation	static	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic
R^3 route labelling	level_1	level_1 level_2	all levels	all levels	all levels	all levels	all levels	all levels	all levels
R^3 path selection	static	static	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic
R^3 stub solution	N/A	N/A	physical level	physical level	physical level	physical level	all level	all level	all level
nonplannar embedding	N/A	N/A	physical links	physical links	all links	all links	all links	all links	all links
node/link failure	N/A	N/A	N/A	physical node	physical node	physical level	all level	all level	all level
Traffic congestion adaptability	N/A	static	level_1	level_1	level_1	level_1	level_1	level_1 level_2	level_1
LNA automation	N/A	static	static	static	static	static	static	static	Quasi-dynamic

Table 3.1 ***Differences between*** R^3 ***versions***

3.3. R^3 OPNET simulation experiments

All our simulations were implemented using the industry-standard network modelling environment, OPNET® Modeler. The general purpose of our simulations is to compare performance of R^3 with existing routing protocols. No flow management or traffic engineering were implemented in these simulations to help R^3 or other protocols reduce packet lost, jitter or out-of-order delivery.

For each of these tests we have conducted a study of goodput, end-to-end delay, end-to-end hop counts, the time evolution of individual buffer occupancies, and out-of-order delivery. The time and link statistics, including mean values of the above quantities, their standard deviations and their maximum values were studied. More details about the settings of these simulations can be found at Appendix E. The figures we discuss and present in this section are only a few examples of these results, plotting the maximum value graphs only for the sake of brevity.

3.3.1. Simulation I: traffic congestion

In these simulations, we compare the performance of R^3 versus that of SPF-based ISIS with ECMP (Equal Cost Multiple Path). Only level 1 loop performance has been measured by R^3. Hop counts are used as 'performance' information to weigh routes or paths at logical levels. In other words, R^3 OPNET modules used in these has been designed to adapt traffic congestion at the physical level and topological changes at all levels.

We have chosen to simulate congestion arising from the creation of a hot spot in a 20-node network approximately based on the core IP network of a European Service Provider. The network is shown in Fig. 3.1.

We have conducted a series of tests gradually increasing the (scaled) packet generation rate at the 20 nodes from 50 to 77, 100, 143, 167 and 200 packets per second. For each of these tests we have conducted a study of goodput, end-to-end delay, end-to-end hop counts, the time evolution of individual buffer occupancies, and out-of-order delivery. The time and link statistics, including mean values of the above quantities, their standard deviations and their maximum values were studied. We discuss and present in Fig. 3.2-3.11 a subset of these results, plotting the maximum value graphs only for the sake of brevity.

For a scaled packet generation rate of 50 packets per second per node, which corresponds to a lightly loaded network, Fig. 3.2 -3.4 show that the performance of R^3 is comparable in most respects to ECMP IS-IS. Unsurprisingly, R^3 packets traverse paths with slightly

longer hop counts and suffer proportionately longer end-to-end delays compared to ECMP IS-IS. What is counter-intuitive though is that the out-of-order delivered packets are fewer in R^3 compared to ECMP IS-IS, as there is little need to adapt in a lightly loaded network provided the initial loading of the network is reasonably balanced (see Fig. 3.3 - 3.4).

Similar results are observed in Fig. 3.5-3.7 for a moderately loaded network with a scaled packet generation rates of 100 packets per second per node, even though the onset of congestion in two of a number of the links of the hotspot node (15) is clearly visible in the case of ECMP IS-IS in Fig. 3.7. In this instance R^3 is outperforming ECMP IS-IS in almost all respects, but only slightly.

Under quite heavy network loading conditions (143 packets per second per node), we can see from Fig. 3.8-3.11 that the ECMP IS-IS network has effectively become congested, while the R^3 network is still operating well, by managing to spread its loading over all available paths.

It is worth remarking that node 15 has 5 links each of 3 Mbps capacity connecting it to the rest of the network. At a network loading of 143 packets per second per node, there is an actual load of 9 Mbps routed towards node 15, nearly all of which are successfully delivered by R^3, whereas only about 82% of these packets are successfully delivered by IS-IS.

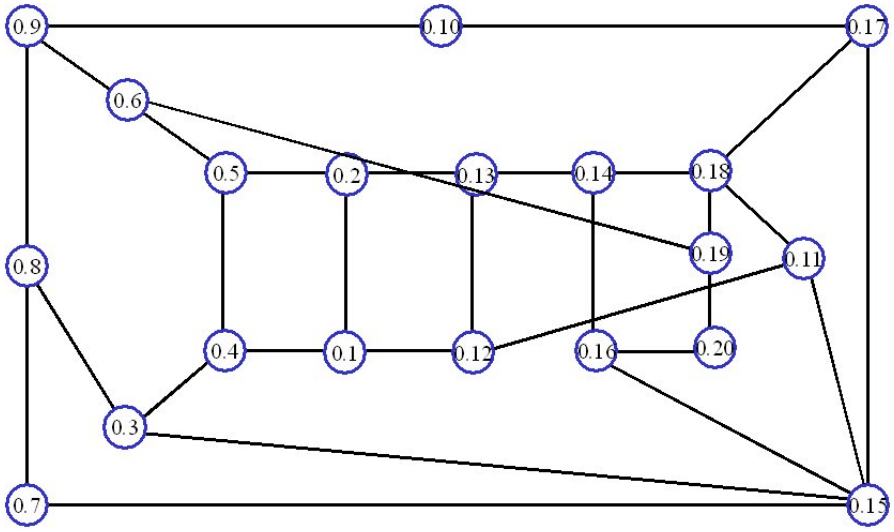


Fig. 3.1: *The network used in the simulation I*

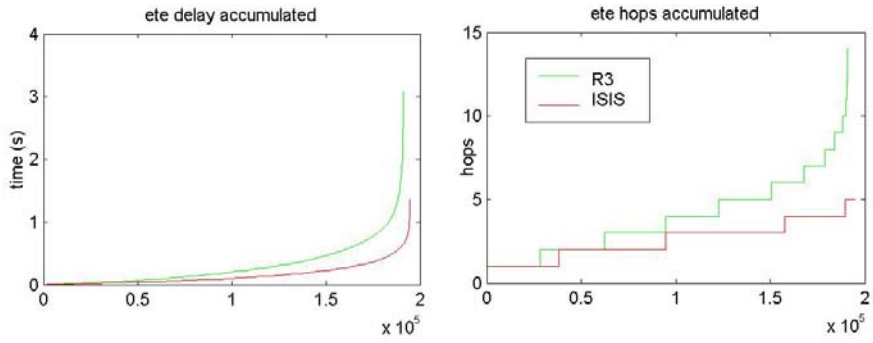


Fig. 3.2: *Comparative 'cumulative' end-to-end-delay analysis of R^3 and IS-IS for a packet generation rate of 50 packets per second (all quantities are scaled); The delay and hop count are sorted in ascending order by packet number*

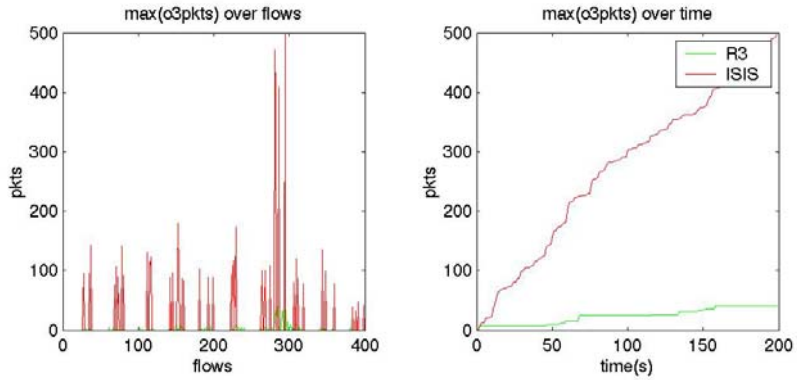


Fig. 3.3: *Comparative out of order delivery analysis of R^3 and IS-IS with equal cost multipath for a packet generation rate of 50 packets per second (all quantities are scaled)*

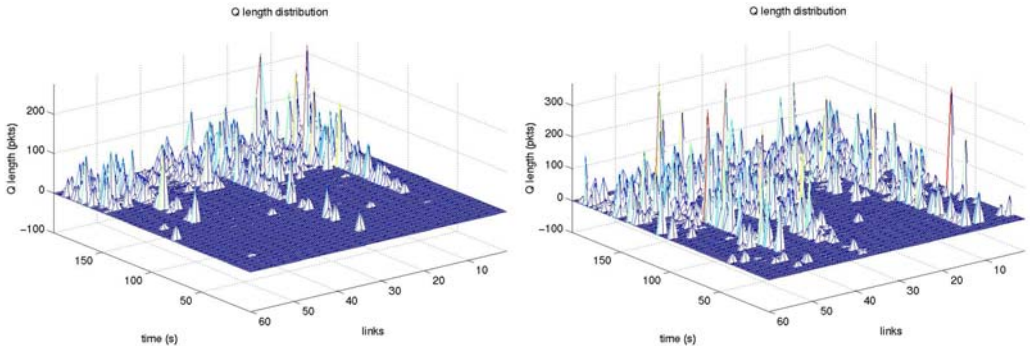


Fig. 3.4: Queue buffer occupancy *of IS-IS (left) and R^3 (right) with equal cost multipath for a packet generation rate of 50 packets per second (all quantities are scaled)*

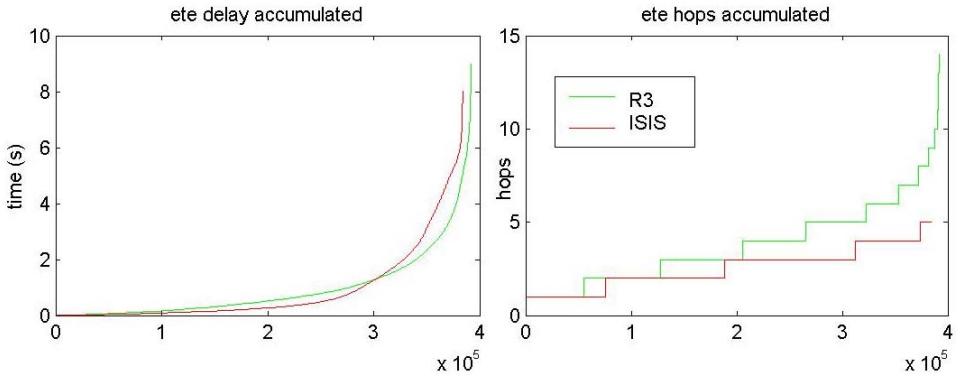


Fig. 3.5: *Comparative 'cumulative' end-to-end-delay analysis of R^3 and IS-IS for a packet generation rate of 100 packets per second (all quantities are scaled); The delay and hop count are sorted in ascending order by packet number*

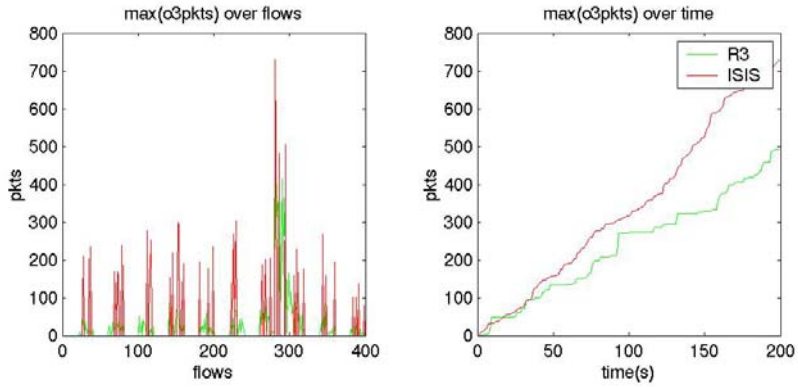


Fig. 3.6: *Comparative out of order delivery analysis of R^3 and IS-IS with equal cost multipath for a packet generation rate of 100 packets per second (all quantities are scaled)*

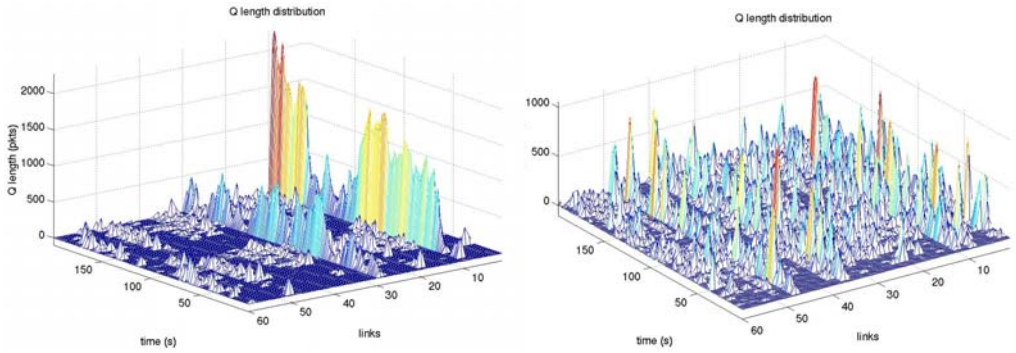


Fig. 3.7: Queue buffer occupancy *of IS-IS (left) and R^3 (right) with equal cost multipath for a packet generation rate of 100 packets per second (all quantities are scaled)*

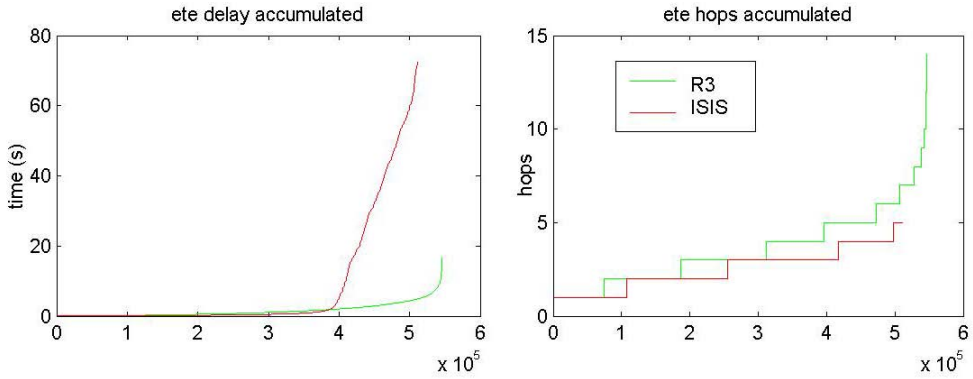


Fig. 3.8: *Comparative 'cumulative' end-to-end-delay analysis of R^3 and IS-IS for a packet generation rate of 143 packets per second (all quantities are scaled); The delay and hop count are sorted in ascending order by packet number*

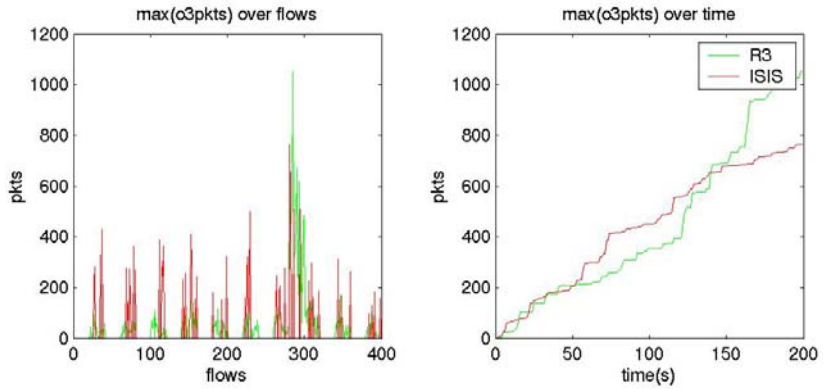


Fig. 3.9: *Comparative out of order delivery analysis of R^3 and IS-IS with equal cost multipath for a packet generation rate of 143 packets per second (all quantities are scaled)*

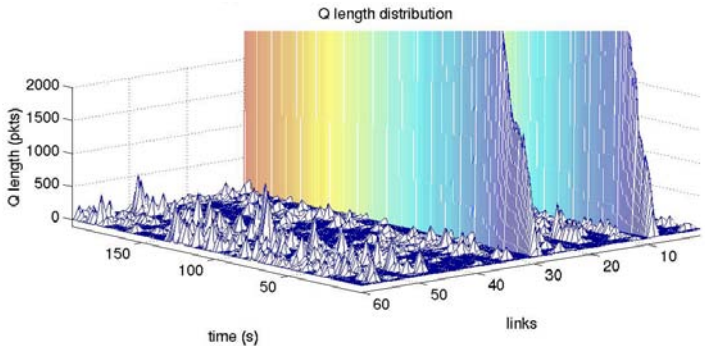


Fig. 3.10: Queue buffer occupancy *of IS-IS with equal cost multipath for a packet generation rate of 143 packets per second (all quantities are scaled)*

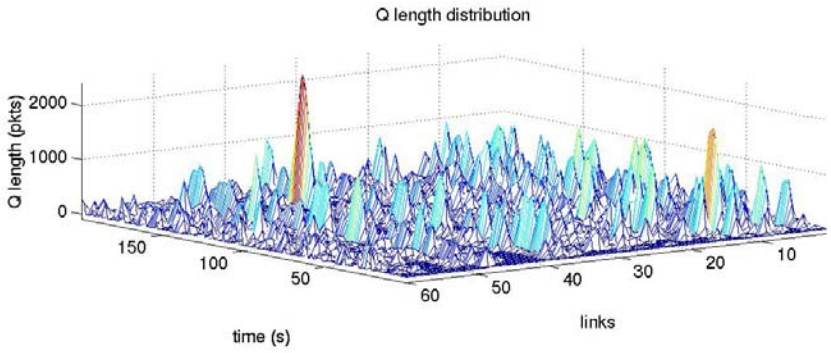


Fig. 3.11: Queue buffer occupancy *of R^3 for a packet generation rate of 143 packets per second (all quantities are scaled)*

3.3.2. Simulation II and III: Scale free networks

Scale free networks arise naturally in many contexts, including the Internet, when new nodes attach themselves preferentially to existing highly connected nodes. We generate a scale free network by using Albert-Barabasi algorithm, where most nodes have the same low degree, but a small number of nodes have the much higher degree. Consequently, it has the advantage of providing highly efficient communication through small number of key, highly-connected nodes. We will assume for the purpose of this study that this statement is also pertinent to military operational communications, with the highly connected nodes representing command and control centres.

As we know, scale free networks are quite robust to random node failures. However, when these highly connected nodes are preferentially targeted (i.e. asymmetric attack), scale free networks can easily be brought down.

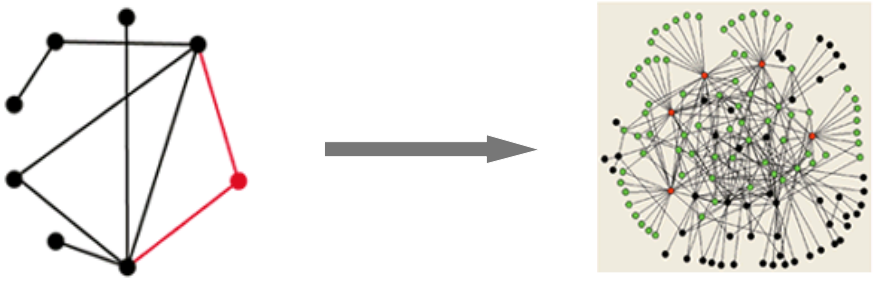


Fig. 3.12: *Using the Albert-Barabasi algorithm to grow a scale-free network*

3.3.2.1. Simulation II: single node failure at scale free network I

To simulate asymmetric attacks to a military battlefield network, we have created an instant failure at a highly connected node in a scale-free network.

We have chosen the data traffic destination distribution to be proportional to the destination node degree. In this case, most of user packets are sent towards highly connected nodes, just as they would be the case of a tactical network where most of the information is transmitted to C&C centers. Compared with nodes of low degrees, these highly connect nodes have more possibilities to become hot spots and/or be physically attacked (i.e. the network can be subjected to an asymmetric attack). In this simulation, we have proved that:

a) R^3 can provide more robust routing than ISIS even though scale-free networks are vulnerable to asymmetric attacks;
b) R^3 can be used as a visualization tool to help improve the vulnerability of scale-free networks.

The 40-node, 54-link scale-free network I and its R^3 levels of abstraction are shown in Fig. 3.13-3.14.

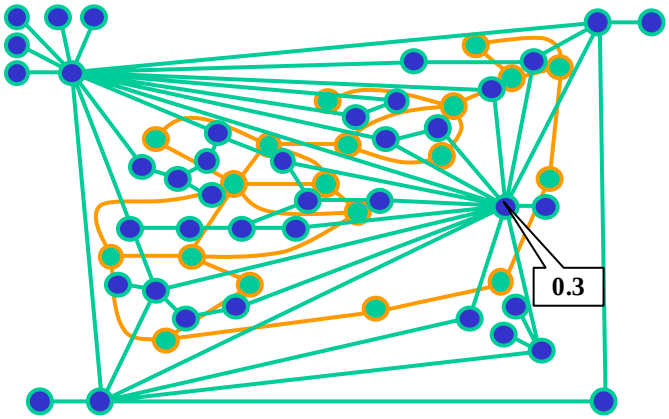


Fig. 3.13: *Levels 1 and 2 of the scale-free network I used in the simulation II with node 0.3 highlighted*

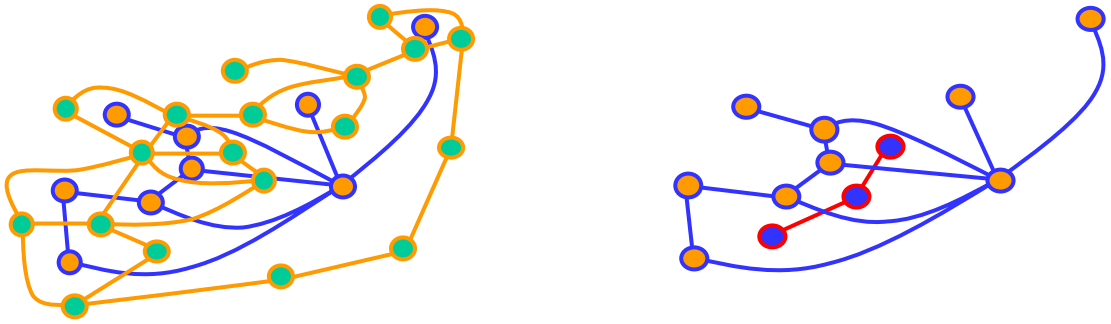


Fig. 3.14: *Levels 2/3 and 3/4 of the scale-free network I of Fig. 3.13*

In Simulation II, we have run a series of tests with gradually increased packet generation rate. As we have learned from Simulation I, there is little need to adapt in a lightly loaded network; also traffic can be too heavy to be adapted for a network with limited capacity. As we can see in Fig. 3.13, when node 0.3 fails, 15 of the total 54 links will be removed from the network and one node is cut off completely. Obviously, the failure of a highly connected node in a scale-free network decreases the network capacity significantly.

Here, we discuss and present comparative performance results (in Figs. 3.15-3.17) at a packet generation rate of 167 packets per second, which corresponds to a reasonably heavily loaded, but not congested, network before and after the node failure.

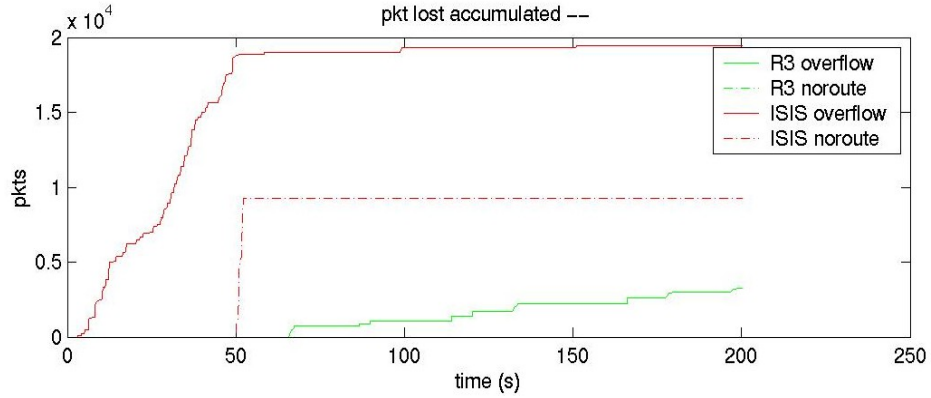


Fig. 3.15: *Comparative lost packet analysis of R^3 and IS-IS with equal cost multipath for a packet generation rate of 167 packets per second (all quantities are scaled)*

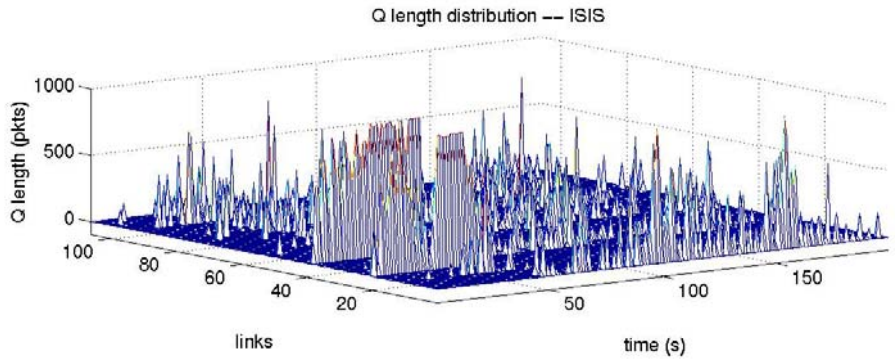


Fig. 3.16: Queue buffer occupancy *of IS-IS with equal cost multipath for a packet generation rate of 167 packets per second (all quantities are scaled)*

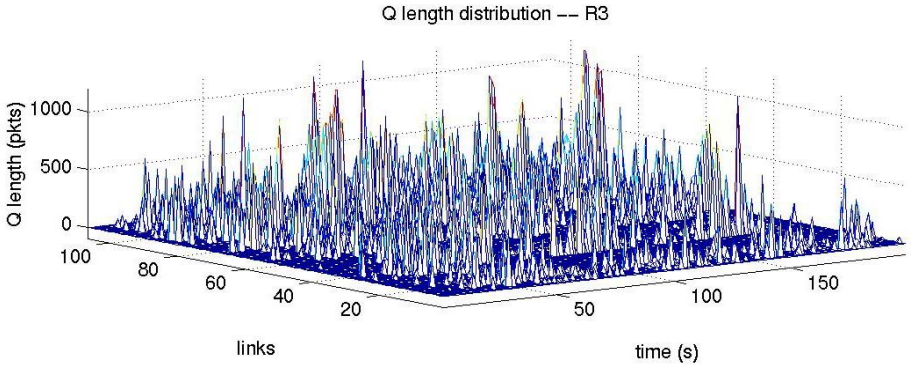


Fig. 3.17: Queue buffer occupancy *of R^3 for a packet generation rate of 167 packets per second (all quantities are scaled)*

Given the equal link weights, a highly connected node in a scale-free network is much more likely to become a critical member of most other nodes' Shortest Path Tree (SPT). The highly connected node 0.3 is such an example. It has become a hot spot in IS-IS since the start of the simulation. Lots of user traffic traverses node 0.3 and causes a large amount of packet loss due to buffer overflows there. After node 0.3 fails at simulation time the 50^{th} second, IS-IS re-converges every node's SPT with associated packet losses due to Dijkstra's non-negligible re-convergence delay. After the node failure, R^3 will also loose packets due to significantly decreased network capacity and available path diversity (i.e. small number of physical and logical rings), as shown in Fig. 3.18.

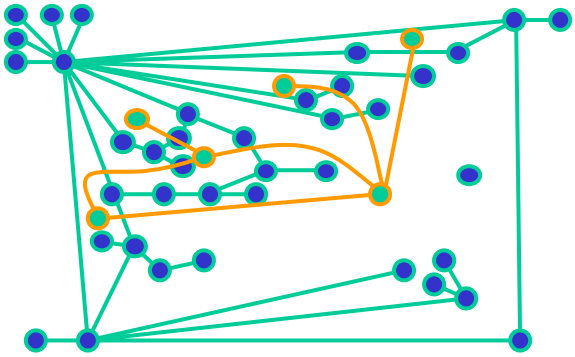


Fig. 3.18: *Levels 1 and 2 of the scale-free network I of Fig. 3.28 after the node 0.3 failed*

By using the R^3 visualization tool (see more details in §2), we modified the scale-free network by only moving the connections of 9 links (as shown in Fig. 3.19), in order to improve the network available path diversity, especially after the node failure. The modified network and its R^3 levels of abstraction are shown in Fig. 3.20-3.21.

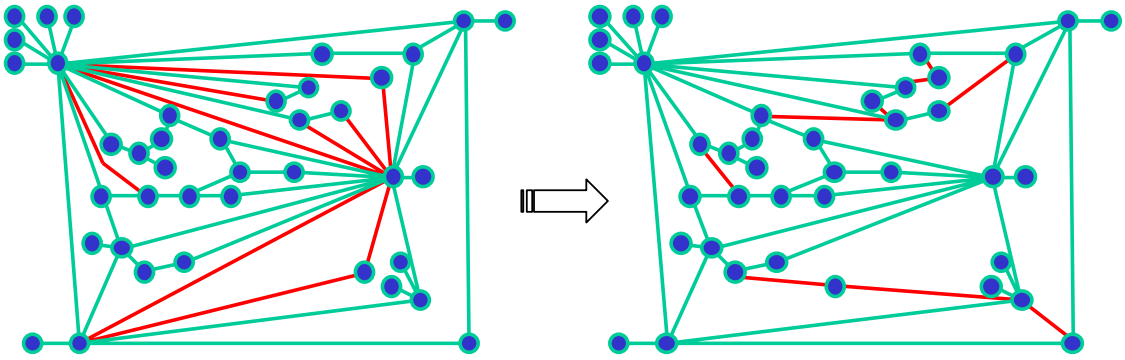


Fig. 3.19: *Modifying the network of Fig. 3.13 to increase path diversity*

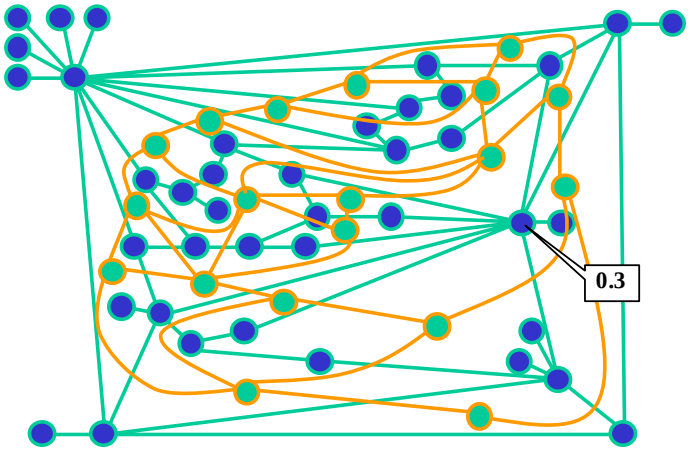


Fig. 3.20: *Levels 1 and 2 of the modified network used in the simulation II with node 0.3 highlighted*

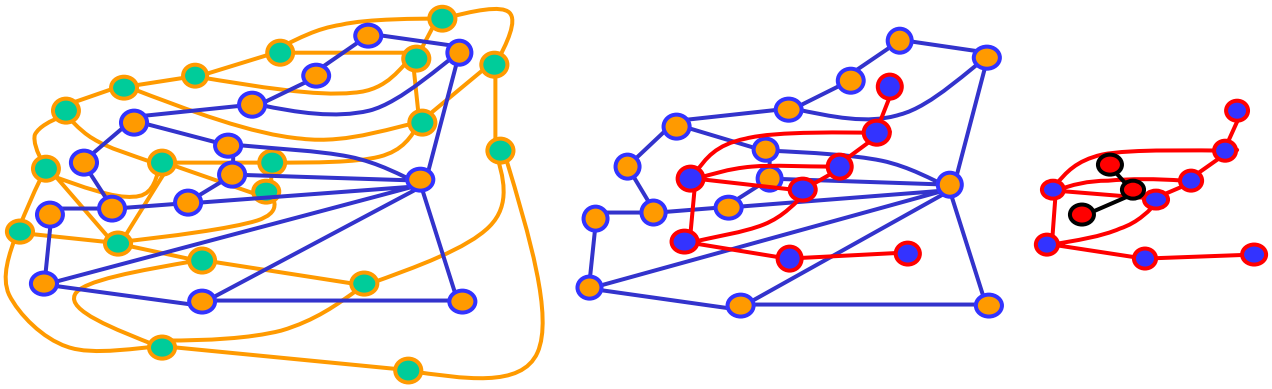


Fig. 3.21: *Levels 2/3, 3/4 and 4/5 of the modified network of Fig. 3.20*

Compared with the scale-free network of Fig. 3.13, the modified network has 5 abstraction levels instead of 4, which implies that it possesses a richer physical path diversity. As shown in Fig. 3.22-3.25, by managing to spread the traffic loading over all available paths, R^3 performs very much better than IS-IS (with ECMP). As shown in Fig. 3.25, the distribution of queue buffer occupancy has no significant changes before and after the node failure, which means R^3 did not even need to adapt to the node failure significantly, as it pre-spread its traffic much more appropriately.

In contrast, the IS-IS network has effectively become congested right after the node failure, even though node 0.3 was not a hot spot to start with this time, because of the topology modification. Hot spots and lost packets arise after node 0.3 failed at the 50th simulation second, which means the modified IS-IS network failed to adapt the node failure satisfactorily.

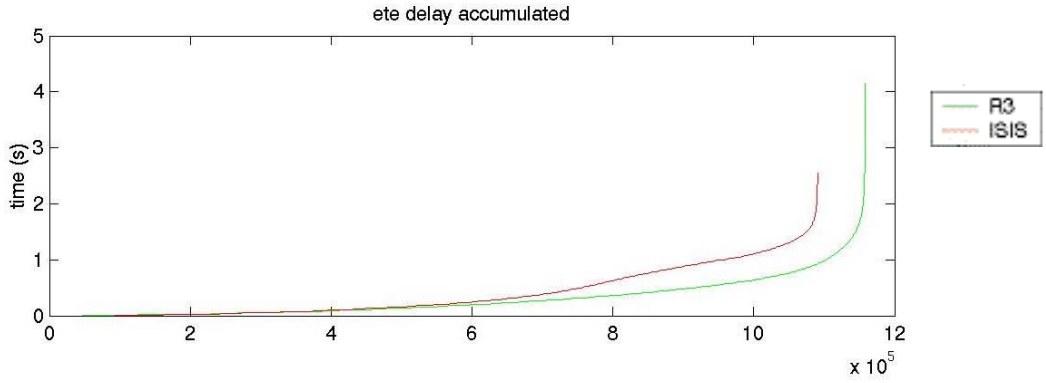


Fig. 3.22: *Comparative 'cumulative' end-to-end-delay analysis of R^3 and IS-IS for a packet generation rate of 167 packets per second (all quantities are scaled) at the modified network of Fig.3.20*

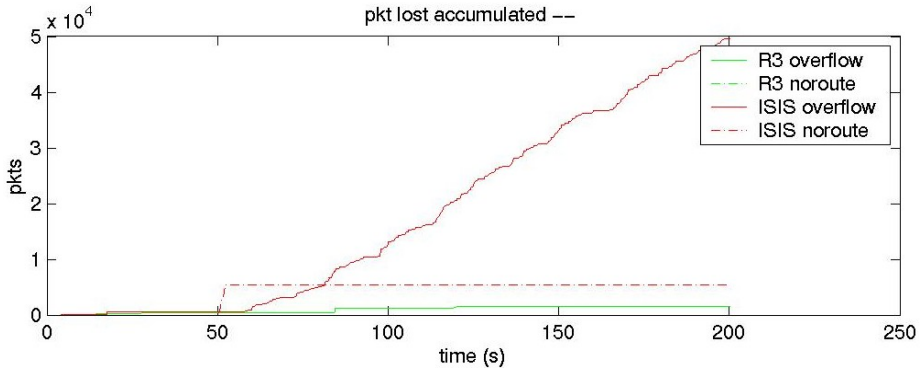


Fig. 3.23: *Comparative lost packet analysis of R^3 and IS-IS with equal cost multipath for a packet generation rate of 167 packets per second (all quantities are scaled) at the modified network of Fig.3.20*

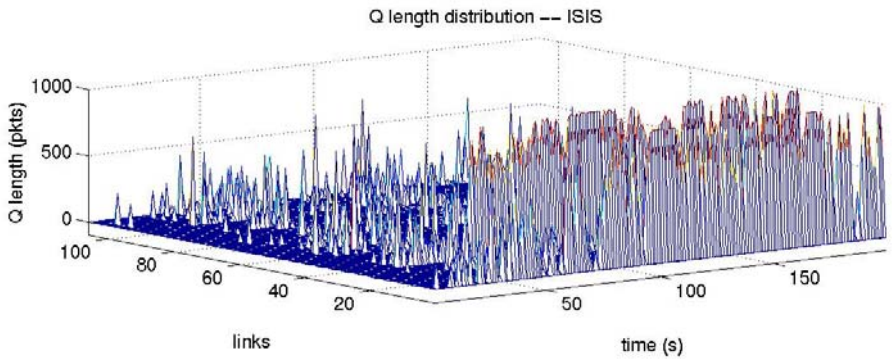


Fig. 3.24: *Queue buffer occupancy of IS-IS with equal cost multipath for a packet generation rate of 167 packets per second (all quantities are scaled) at the modified network of Fig.3.20*

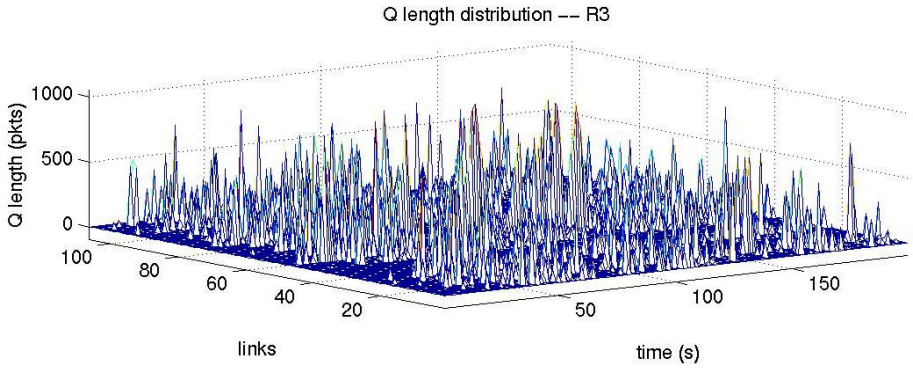


Fig. 3.25: Queue buffer occupancy *of R^3 for a packet generation rate of 167 packets per second (all quantities are scaled) at the modified network of Fig. 3.20*

3.3.2.2. Simulation III: multiple node failures at scale free network II

To simulate a much more severe asymmetric attack on a tactical communication network, we have created multiple failures at highly connected nodes in a scale-free network. Unfortunately, in simulation II, our scale free network I is relatively small and as a result multiple node failures partition the network. A consequence of this is that meaningful performance comparisons on 2-3 partitioned tree-like sub-networks are not possible.

Similarly to simulation II, we have employed the Albert-Barabasi algorithm to create a 120-node, 117-link scale-free network. For simplicity, we planarized this by removing 17 non-planar links and subsequently removing all the stub and purely transient nodes in order to make the simulation time more manageable, without loosing any significant feature in the problem under study. The resulting scale-free network II has 39 nodes and 70 links and its R^3 levels of abstraction are shown in Fig. 3.26 – 3.27.

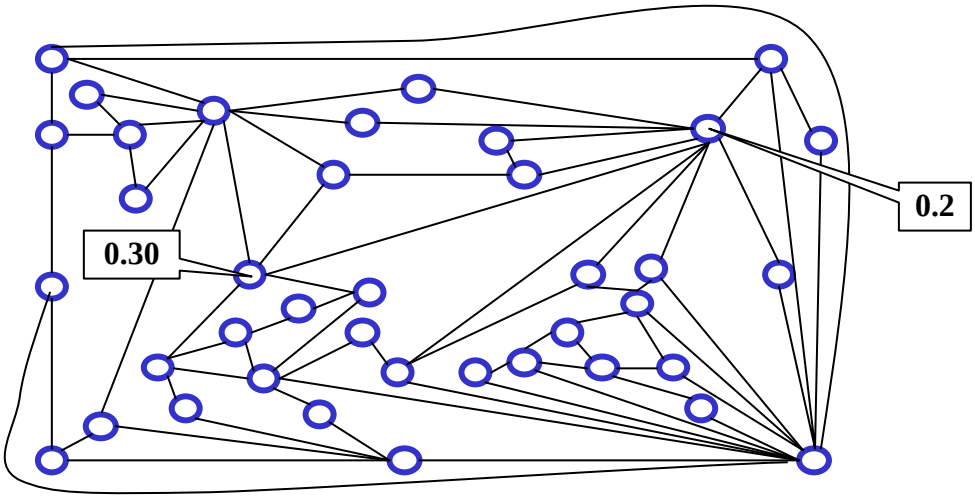


Fig. 3.26: *The scale-free network II used in the simulation III with nodes 0.2 and 0.30 highlighted*

Compared Fig. 3.13 with Fig. 3.26, the scale-free network II has many more links and loops than the scale-free network I, which translates to significantly more underlying path diversity even after multiple node failures. As shown in Fig. 3.27, after node 0.2 and node 0.30 fail, a logical link at level 2 becomes broken. This requires R^3 to adapt to topological changes (e.g. link broken) not only at the physical level (as in Simulation II), but also at the logical levels.

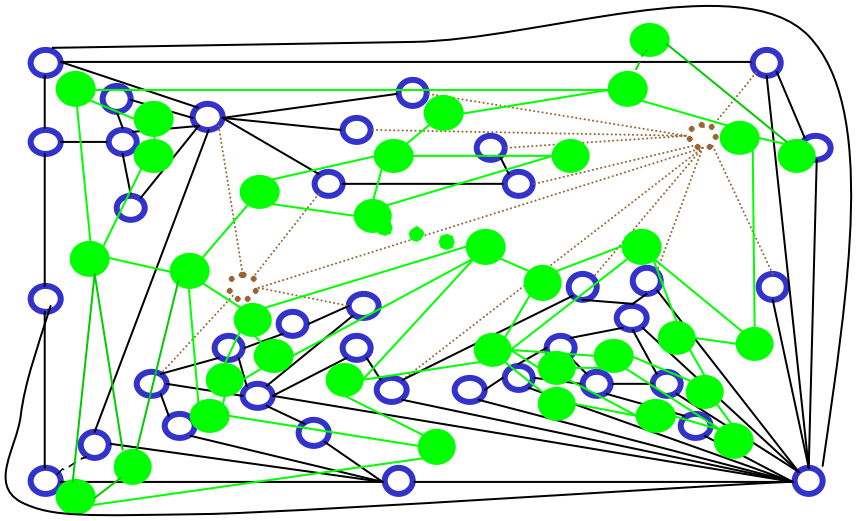


Fig. 3.27: Level 1 and level 2 of t*he scale-free network II used in the simulation III after nodes 0.2 and 0.30 failed*

As shown in Fig. 3.28, the R^3 route levels *in use* in the scale-free network II are overall higher than those in scale-free network I for our choice of end-to-end source-destination pairs. In order to avoid traffic congestion in the scale-free network II, R^3 requires the ability to adapt to traffic changes (e.g. congestion) at the logical levels. In simulation I and II, R^3v4 was employed (see more details about different R^3 versions in §3.2). The traffic congestion adaptability of R^3v4 demonstrated in §3.3.1 and §3.3.2.1 was only at the physical level (i.e. level 1). Specifically, R^3v4 algorithms can only choose level 1 routes dynamically to avoid traffic congestion. These dynamical decisions are made based on the periodical performance measurement on level 1 rings. The decisions made by R^3v4 to choose high-level routes are static, based on higher-level route hop counts. These static decisions will limits the traffic congestion adaptability of R^3, especially when most of the R^3 routes employ the higher logical levels, as is the case in the scale-free network II of simulation III.

As in simulation I and II, we have run a series of tests with gradually increased packet generation rate for Simulation III. Here, we discuss and present the results (in Figs. 3.29-3.30) of the generation rate at 142 packets per second (scaled), which corresponds to a reasonably balanced network loading before and after the node failures (moderately loaded, but not congested).

In simulation III, we can see that R^3v4 does not performs very well compared with R^3v7 and R^3v8, although it still performs better than IS-IS (with ECMP). R^3v4 has not been designed to be able to adapt topological changes, or traffic congestion at a logical level.

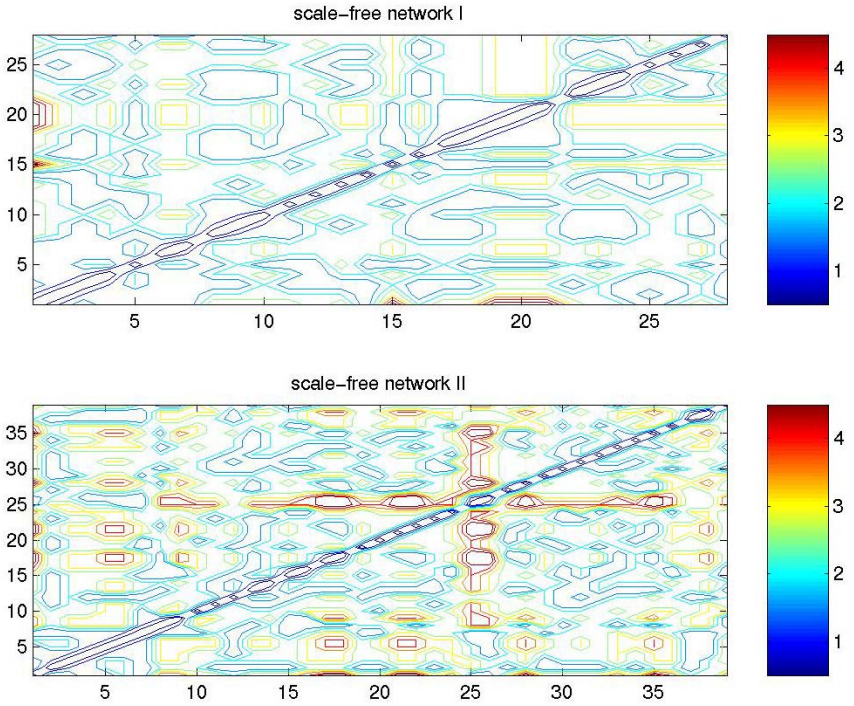


Fig. 3.28: *Comparative route levels of R^3 at scale-free network I and II*

R^3v8 has the best performance, which is designed to adapt to topological changes and traffic changes at all levels. R^3v8-1 is a partial implementation version of R^3v8 that can adapt to topological changes at all levels and traffic changes at levels 1 (i.e. the physical level) and 2 only. R^3v7 has the second best performance, which can only adapt to traffic changes at level 1.

The differences between R^3v7 and R^3v8-1 are limited, but significant. By adaptation at the physical level, R^3v7 can redirect traffic to avoid certain congested or damaged physical links or nodes; while by logical level adaptation, R^3v8-1 can literally balance and shift traffic from one heavily loaded physical area to others. In Fig. 3.30, we can see that only R^3v8-1 is capable of maintaining the maximum buffer occupancies to be low after the second node failure at the 60^{th} simulation second, whereas R^3v7, R^3v4 and IS-IS are all either about to go into congestion or have already done so. This is strong evidence that *in order to built highly resilient networks, not only needs underlying path diversity, but also adaptation at all logical levels.*

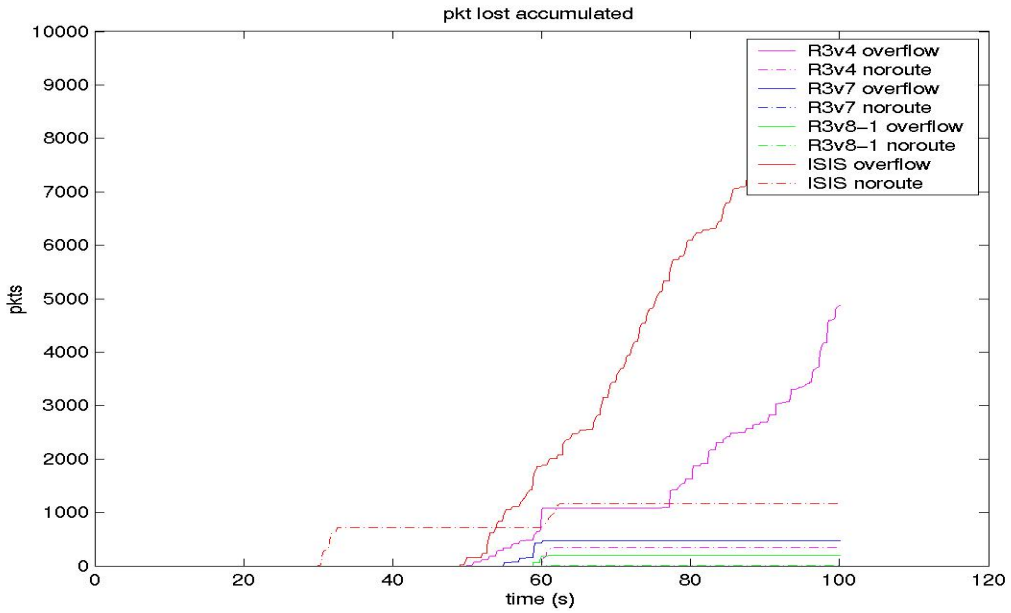


Fig. 3.48: *Comparative lost packet analysis of different R^3 versions and IS-IS with equal cost multipath at the scale-free network II of Fig.3.26*

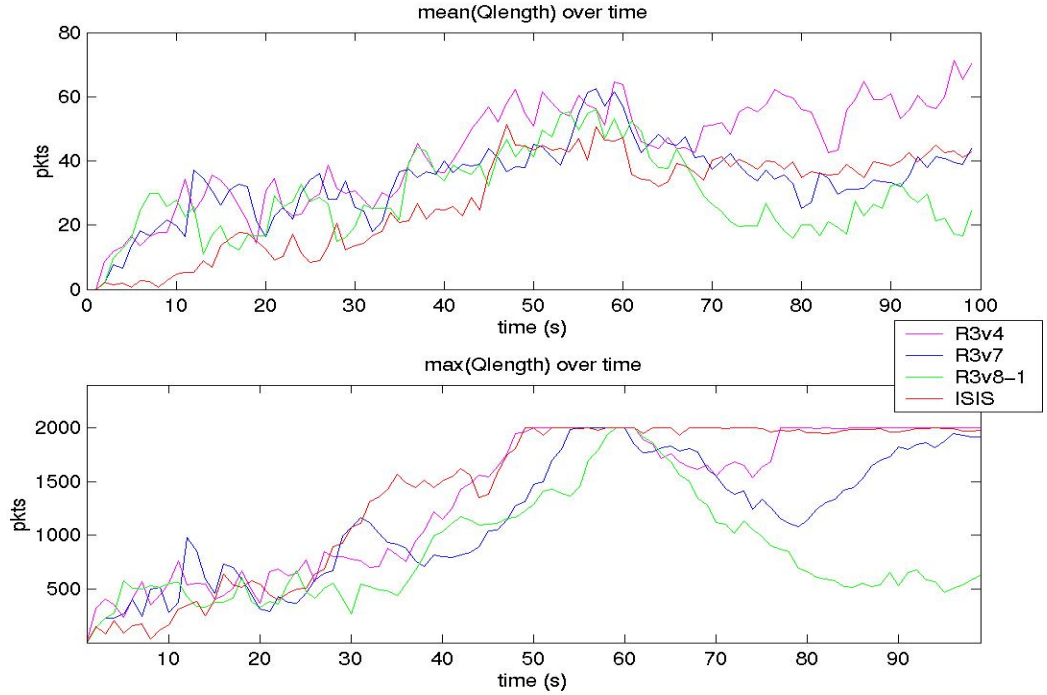


Fig. 3.30: *Comparative queue buffer occupancy analysis of different R^3 versions and IS-IS with equal cost multipath at the scale-free network II of Fig.3.26*

3.3.3. Simulation IV: R^3 mobility simulation

In previous simulation experiments, it was established that R^3 was able to demonstrate significant improvements in performance and resiliency over traditional SPF (Shortest Path First) routing algorithms in situations of congestion and failure. In the second research activity we therefore seek to demonstrate that R^3 will also be able to deliver those benefits whilst sustaining reasonable levels of router node mobility in the network. By successfully demonstrating this capability, it will be possible to extend the networking benefits that R^3 provides to mobile networking scenarios within the theatre of operations.

R^3 v9, as introduced at §3.2, can dynamically generate the LNA architecture from scratch for certain constrained topologies. A single queue node module (see more details at Appendix D) was also introduced to better reflect the shared domain of a wireless network, where every node has to compete against each other to access a common medium.

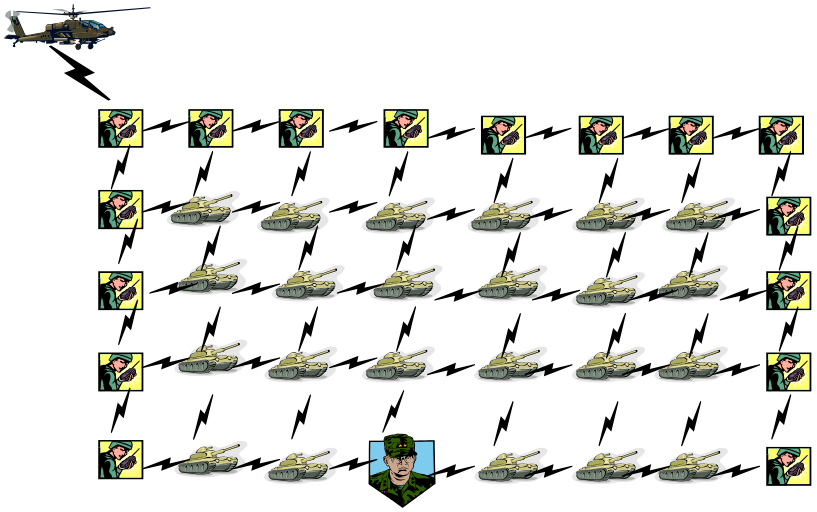


Fig. 3. 31 A simple battle-field network

In Simulation IV, as shown in Fig 3.31, there is a mobile helicopter feeding video information back into an operational network of military positions on the ground. The traffic from the helicopter to the Command and Control centre is therefore considered to be a constant stream of video. Meanwhile, every node on the ground is also sending messages to the C&C centre as background traffic. We have therefore run a series of tests with gradually increased background traffic.

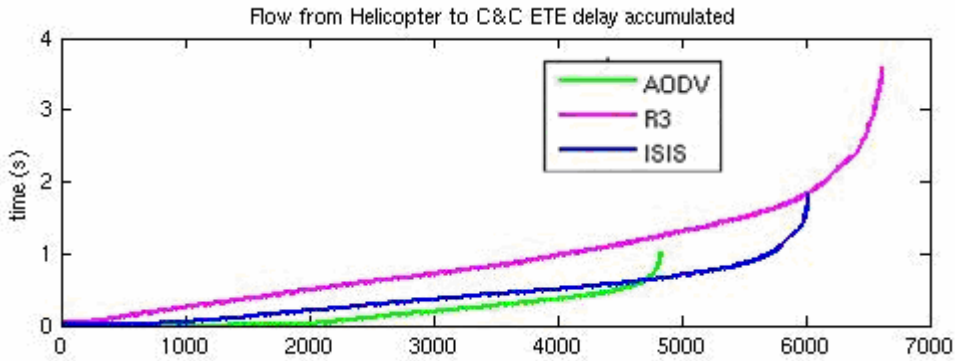


Fig. 3.32 Comparative 'cumulative' end-to-end-delay analysis of R^3, IS-IS and AODV for a modest background traffic at the network of Fig.3.31

As we can see in Fig. 3.32, with a reasonable level of background traffic, both ISIS and AODV have packet lost due to node mobility. The packet lost rate for ISIS and AODV are more than 10%, while R^3 delivers all 6,600 packets from the helicopter to the C&C node successfully. Fig. 3.33 shows sample frames that demonstrate the impact of packet loss rates on video information transmission[10]. Increasing packet loss rates on video streams will not only result in more heavily damaged frames but also in more damaged frames. The original video streams with these packet lost rates can be found at the attached CD.

Fig. 3.33 Frame samples with various packet transmission lost rates at 1%, 10% and 20% respectively

The inherent advantages in R^3 for mobile applications come from its ability to exploit path diversity, to back track data when failures are first discovered and to separate out routing from topology discovery. With an SPF algorithm or AODV, there is either a single path, or a limited number of paths for each route. Hence whenever a node moves, an SPF based or AODV network will usually have to re-converge on a new set of routes in order to stay in communication with that node. An R^3 network with link diversity to

[10] Philippe De Neve, *MPEG video streamed over an IP-based network with packet lost,* www.ibcn.intec.ugent.be/css_design/research/topics/2003/FTW_PhD29_PhilippeDeNeve.pdf

the mobile node will behave quite differently:

- When a link is lost, as a result of the mobility of the node, traffic is simply diverted onto the remaining links without the need for re-convergence.
- Any data that is stranded, as a result of the loss of the link, can be backtracked to a point where it can gain access to an alternative link.
- When new links are formed, as a result of the mobility of the node, the topology of the new network can be discovered without impacting the current forwarding tables. Then at an appropriate time when the new topology is synchronised across the network, routing can be switched over to the new topology with low risk.

The simulation introduced wireless links instabilities with various time-scales to simulate the operational realities of wireless networks. Under these conditions the advantages of R^3 becomes clear. As can be seen, IS-IS exhibits huge delays as one link goes into overload and as a result it is only able to deliver just over 5,000 packets from the helicopter back to C&C. AODV exhibits low delays but is only able to deliver just over 4,000 packets. R^3 on the other hand copes extremely well, delivering all 6,600 packets and showing near equivalent delay performance to AODV.

Comparing Fig. 3.32 and Fig. 3.34, we can see that the performance of R^3 has been hardly affected by the introduced instability, which is important when considering realistic battle-field environments. The ability of R^3 to simultaneously exploit multiple paths through the network enables it to cope with both modest levels of mobility as well as preventing congestion. This is illustrated by the queue lengths across the network at Fig. 3.35, which shows how R^3 (on the right) is able to distribute the traffic across the network compared with IS-IS (on the left).

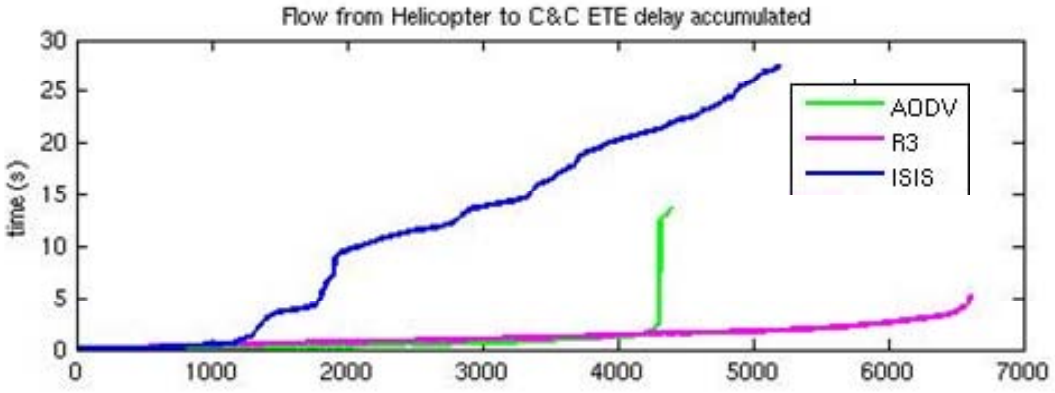


Fig. 3.34 Comparative 'cumulative' end-to-end-delay analysis of R^3, IS-IS and AODV for the same network as Fig.3.32 with additional wireless interference introduced

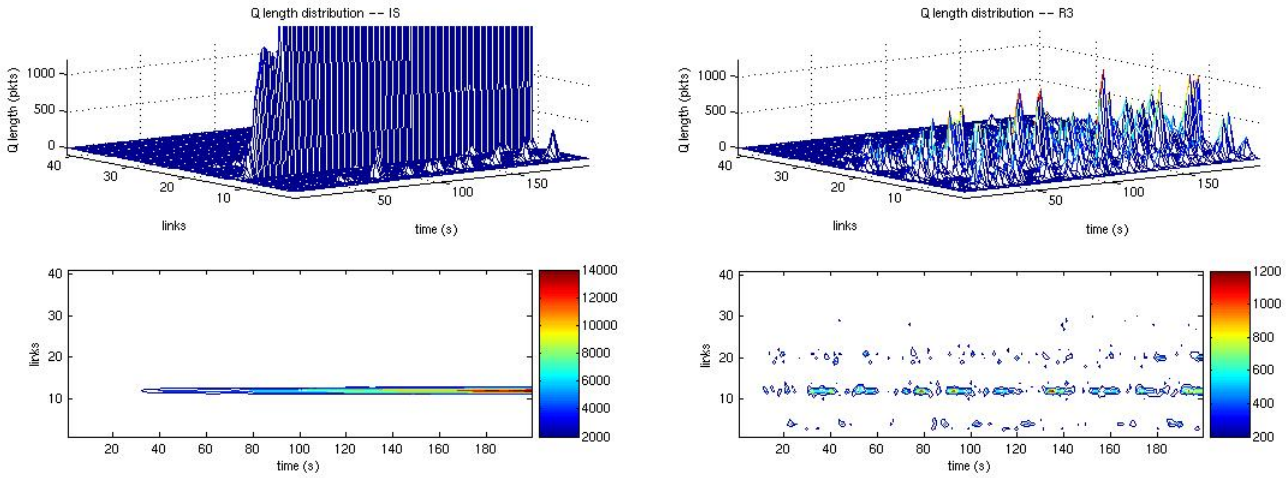


Fig. 3.35 Queue buffer occupancy of ISIS and R^3 for the scenario of Fig. 3.33

4. Conclusion and Recommendations

An innovative new approach to achieve robust high performance networking in packet switched networks has been demonstrated in the two research activities funded by AFOSR EOARD. The approach is based on a Logical Network Abridgement (LNA) that quantifies the diversity of paths in a network without resorting to clustering or the imposition of arbitrary additional criteria. The LNA process identifies an independent basis set of cycles and abstracts these recursively.

The research activities have shown that a dynamic routing protocol called R^3 (Resilient Recursive Routing) designed to exploit the LNA abstraction has significantly improved performance and robustness over conventional fixed and mobile shortest path first (SPF) routing algorithms in situations of congestion, attrition and limited node mobility.

The first research activity has also shown, through the development of a demonstrator, how the LNA abstraction can also be exploited as a visualisation tool that enables operational staff to more effectively assess and manage network infrastructures. The visualisation tool can be used in conjunction with any dynamic routing protocol.

Based on this encouraging progress, we would recommend two options for taking this work forward into 6.2 funded lab prototyping activities.

Routing Prototype:

There are three inter-related fundamental questions which must be addressed in order to create a prototype routing technology based on these concepts:.

- The pre-requisites to creating a fully-automated, fully-distributed R^3 protocol are to enable the full automation of a distributed LNA algorithm, and to provide a globally-consistent naming algorithm for all levels of abstraction. This activity will entail the verification of candidate algorithms and their incorporation into a fully functional lab prototype implementation of R^3.

- An important further advancement is to achieve stable dynamic routing in such an environment based on a mixture of measurements, policies and metrics. The aim of this research will be to add this capability to a lab prototype environment of wired and/or wireless links with limited node mobility and highly inhomogeneous link capacities and delays. This activity will culminate in defining constraints on the measurements, policies, metrics parameters and adaptation time-scales that result in a protocol free from routing instabilities.

- An important issue in networking, irrespective of whether we consider large-scale wired or mobile ad hoc wireless networks, is to ensure stable routing protocol operation with incomplete, dynamic topology discovery and thus relaxed requirements for globally consistent topology information. The underlying challenge is how to build the LNA abstraction from the bottom up (i.e. constructing a local minimum cycle base) and reconciling these abstractions for inconsistencies and naming differences in adjacent overlapping regions, at all different logical levels. This activity will seek to enhance the lab prototype so that it is tolerant to such inconsistencies and to understand the limits that continue to ensure proper operation.

Visualization Tool Prototype:

- A working visualization tool also requires the full automation of the algorithms referred to the routing paragraph above.
- The inverse abstraction problem that is necessary in order to develop and automate operational decision support algorithms for the network operator (i.e. the "what if" suggestions) is an extremely challenging problem that will require a substantial research effort.
- Prototyping such a tool and improving its user interface and usability will also require substantial work.
- The LNA abstraction within the tool could also be combined with various network performance metrics and their time-rates of change at the various levels of abstraction in order to monitor network activity (as opposed to performance) and could provide a fresh approach to the early detection of undesirable traffic (e.g. worms) on the network.

Appendix A: An Alternative LNA Application Example

LNA has potential applications in many fields, for example constructing resilient routing protocols for complex communication networks; informing the best way of containing the outbreak of diseases; analysing social networks; or guiding planning decisions to avoid congestion prone transport networks.

An example application area is the spread of epidemic diseases and their control. In this case, it is important to determine the impact of a particular node to the spreading of a disease, i.e. what would happen if a node were to be eliminated as a consequence of a successful immunisation programme or quarantine imposition? A simple but useful index which indicates the effect of the removal of a node on both the connectivity and path diversity of a network is the vulnerability index of node i, Vi, defined as, $Vi \equiv (Nbefore/Nafter)\times((Lbefore + 1)/(Lafter + 1))$, where Nbefore, Nafter are the size of the biggest connected cluster in the network with node i present and removed, respectively, and Lbefore, Lafter are the corresponding maximum numbers of LNA abstraction levels. The LNA can be further exploited within this application area, by analysing the coarse-grained connectivity between densely connected clusters/communities and assist in prioritising the treatment or containment of a disease to prevent an epidemic from becoming a pandemic. Clearly, the LNA is not a suitable analysis tool in itself, but needs to be combined with existing techniques that describe infection dynamics in networks[11]. For example, the disease transmission probability for each physical link needs to be translated to a corresponding probability for higher-level logical links according to some (here unspecified) rules that are meaningful epidemiologically. The essence of the translation of probabilities along a logical link at the next higher level is to compute the average cycle-to-cycle infection probability via all possible paths through their common nodes and assign this to the logical link.

Insights into social networks can also be gained through the application of the LNA analysis. Figure below shows the result of the application of the clustering technique proposed by Newman and Girvan[12] and the corresponding analysis using the LNA to compare with their result. It can be seen from the figure below that the logical level 2 gives rise to three disconnected network clusters in agreement with Newman and Girvan's work. However, the membership of these clusters, also shown in the figure here, has differences and a physical node can belong to more than one cluster, acting as a link between communities. These nodes can be viewed as being very important in the spreading of gossip or diseases in such a networked community. The clustering arising from the LNA procedure does not use any criteria extrinsic to the network, but is a natural consequence of the available physical connectivity.

[11] R. M. May, A. L. Lloyd, *Phys. Rev. E* **64**, 066112 (2001)
[12] M. E. J. Newman, M. Girvan, *Phys. Rev. E* **69**, 026113 (2004)

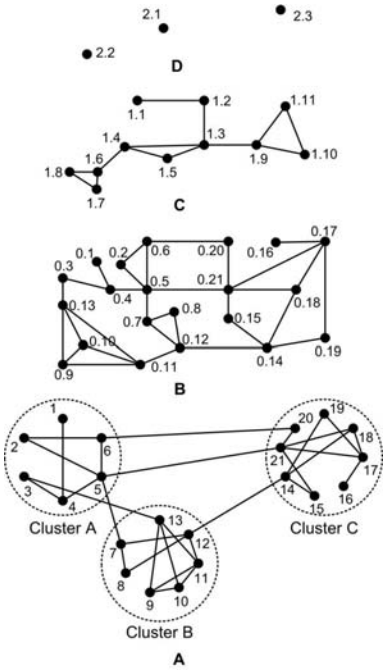


Figure (A) The network clustering example of Fig. 1 of (*9*), where the clustering analysis yielded cluster A′ = {1, 2, 3, 4, 5, 6}, cluster B′ = {7, 8, 9, 10, 11, 12, 13} and cluster C′ = {14, 15, 16, 17, 18, 19, 20, 21}. (B) A planar embedding of level 0 of the same network. (C) Logical level 1 of the network. (D) Logical level 2 of the network having three disjoint clusters, whose membership can be mapped iteratively down to physical nodes as, cluster A = {1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 20, 21}, cluster B = {9, 10, 11, 13} and cluster C = {14, 15, 16, 17, 18, 19, 21}. Nodes 11 and 13 act as gateways between clusters A and B as they occur in both sets, whereas nodes 14, and 21 (but not 15 as it is a transient node) act as gateways between clusters A and C. Clusters B and C are not connected to each other as is evident from logical level 1.

Appendix B: A Comparison of SPF and MPLS with R^3

B.1 What is wrong with today's network protocols?

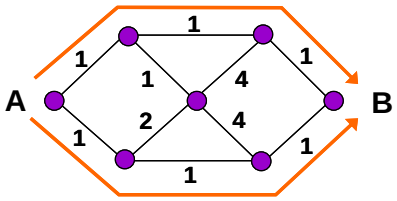
B.1.1 Dynamic routing (based on SPF)

Today's dynamic routing algorithms can be classified as:

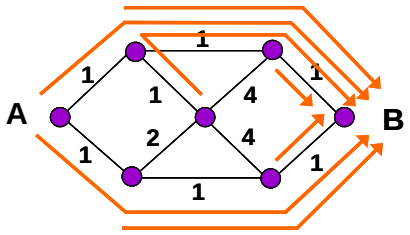
- Proactive routing, where routes are explored regardless of routing request, which can be further classified as link-state based (such as IS-IS and OSPF) or distance-vector based (such as RIP).
- Reactive routing, where routes are only explored on demand, such as AODV and other Mobile ad hoc routing algorithms.

Despite of their significant differences in many terms, IS-IS, OSPF, RIP, AODV and most of existing routing protocols only use a small sub-set of all available paths for routing. The most popular method to calculate such a preferred sub-set is Professor E.W. Dijkstra's SPF algorithm created in 1956.

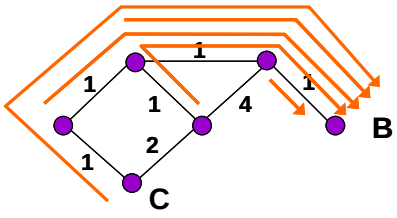
The figure below shows a very simple network that can be used to illustrate how Dijkstra's SPF algorithm enables Router A to establish the shortest path across the network to Router B, based on the cost associated with each link in the network. In the case of the example, the shortest paths for a route from A to B are both ways around the perimeter of the network, and if the router supports Equal Cost Multi Path (ECMP) routing then it can be seen that Router A will have the choice of sending data over either path to B – providing a measure of resiliency.



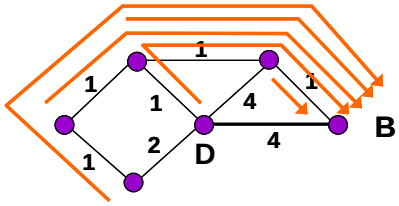
Dijkstra's SPF algorithm was a huge step forward when it was first introduced into routing protocols, but its approach optimizes the selection of each path across the network in isolation and does not take into account the aggregated impact on the network. As a result, it is very often the case that "hot spots" will occur in the network as the same low cost segments are selected as the shortest paths for multiple routes. This is illustrated in the figure below by considering all the routes into Router B for our simple network.



It is quite apparent that the overall resources of the network are not used to good effect in delivering the traffic to B, and that it is only on the route from A to B that there are two alternative equal cost paths. Therefore, if part of the network is lost, see figure below, Router C must wait until the impact of this loss is flooded through the network and it re-converges on a new set of paths.



In addition, it is even possible for additional resources to be added into the network, but for it to have no impact on network congestion as a result of the cost that has been associated with it. This is shown in the figure below, where an additional link has been added between D and B, but the high cost associated with it means that it does not become the preferred path for any of the routes into B.



Advantages:
Dynamic routing protocols based on SPF provide a basic level of robust networking, as they are good at maintaining a best-effort level of connectivity on all routes across the network.

Workarounds:
It is of course possible to improve the overall performance of the network, by "tuning" the costs associated with each of the links in order to better the balance the traffic load against the available network resources. Such an approach though is only easily applied to commercial networks, which tend to be stable and over engineered. Even in these more forgiving circumstances such changes will often result in the hot spot moving to a new

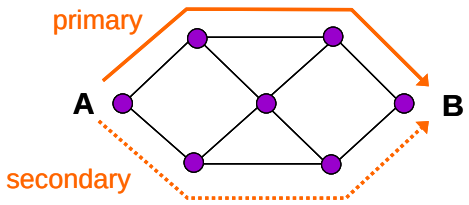
location in the network rather than being eliminated. Prioritization is also being increasingly used to expedite important packets through congested links in order to provide specific traffic types with an appropriate QoS. However, by also making better use of the overall resources of the network then it is possible to also minimize the impact of such prioritization on the other traffic types.

Disadvantages:
Unless the network is built to very strict architectural guidelines, it is also unlike that there will be many alternative equal cost paths through the network. Therefore network failures will often cause a temporary loss of many routes, which will only be recovered after the impacts are flooded through the network enabling it to re-converge on a new set of paths. Large amounts of the network's resources can also be severely under-utilized, which can be particularly frustrating if they could be carrying traffic that would relieve congestion in other parts of the network.

B.1.2 MPLS (Multi Protocol Label Switching)

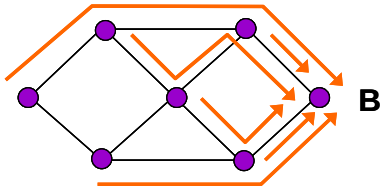
In the past, the "best effort" service of a traditional dynamically routed network has typically been sufficient to meet the demands of the networked applications that they supported. However, this is changing rapidly in the commercial world as converged IP networks look to support more demanding services, such as voice and video. NCW is now bringing the same demanding requirements to military networks, but with the added challenge of addressing them in a highly dynamic and unpredictable context.

MPLS has become the recognized solution in the commercial world for addressing some of the disadvantages found with traditional dynamic routing. MPLS's strength is its connection-oriented nature. It allows paths to be set up across a network, and for data packets to be associated with those paths through the addition of a label to each packet. Furthermore, to cope with unexpected network failures, two paths can be established on each route: a primary path that is usually used and a secondary path that can be used in the case of a failure of the primary path. The figure below provides an example of primary and secondary MPLS paths being established between A & B.



As MPLS establishes a specific connection across the network, resources can be reserved against that connection in order to provide a guaranteed QoS. The use of connections, with resource reservation, can now be used to improve the distribution of the traffic

across the network – ensuring that the network is capable of supporting the predicted traffic loads. This is shown in the figure below for all of the traffic to B.



MPLS certainly provides one approach to overcoming the disadvantages of dynamic routing, but it does that at the expense of moving from a connectionless to a connection-oriented environment. Additional synchronized state information must be held across the network for every connection, and if the environment is one of rapid change then these connections must be torn down and re-established frequently. The scale of that challenge becomes apparent as the number of nodes increases:

- With 7 nodes: each has 6 primary and 6 secondary paths total paths = 7 x (2 x 6) = **84**
- With 700 nodes: each node has 699 primary and 699 secondary paths total paths = 700 x (2 x 699) = **978,000**
- With 70,000 nodes: each node has 69,999 primary and 69,999 secondary paths total paths = 70,000 x (2 x 69,999) = **9,799,860,000 (lets say 9.8 billion)**

Advantages:
There is no doubt that people often feel more confident with connections than with relying on random processes. Connections allow the resources of the network to be directly controlled, and for resources to be reserved against a connection so that guarantees of its performance can be given. Such approaches work well in stable environments for:

- Very large numbers of identical connections (e.g. the telephone network);
- Medium numbers of slow changing hierarchical connections (e.g. SDH).

A connection-oriented paradigm also allows diverse primary and secondary paths to be established to enable restoration in the event of isolated failures.

Disadvantages:
As was shown earlier, if all connections are established in advance, then we can quickly hit issues of scale. However, if connections are established on demand then we have to wait for them to be set up before we can use them, and wait for them to be torn down before their resources are released for reuse. The introduction of connections has significantly increased the amount of synchronized state information that must be held across the network. This is difficult enough to achieve in a stable environment, let alone one that is unpredictable and rapidly changing. Finally, it is also important to note that

although resources can be assigned to a connection, prioritization will still be required to differentiate the QoS of different types of traffic within that connection. This could be resolved by creating multiple connections for the different traffic types, but this would only compound the scaling issues.

B.2 The opportunity for R^3 – dynamic engineering

Existing networking options therefore provide us with a stark choice:

- Dynamic Routing that is robust and scalable, but is centred on establishing best effort connectivity, and
- MPLS Traffic Engineering that provides control but lacks scalability and survivability in a rapidly changing environment.

R^3 provides a fundamentally new approach that enables the traffic load offered to the network to be dynamically mapped to the available network resources – Dynamic Engineering. Routes across the network are defined in terms of a recursive abstraction of the network's topology. This creates two key advantages:

- Flexibility: as a deterministic route at a higher layer in the abstraction can be realised in a number of different ways in the lower levels of the abstraction. This provides each route across the network with multiple paths, rather than a single path as is usually the case with dynamic routing.
- Scalability: as the abstraction provides a hierarchical structure of connections based on the topology of the network that can be dynamically combined in order to build each of the required paths across the network, rather than having a separate connection for each separate path as with MPLS.

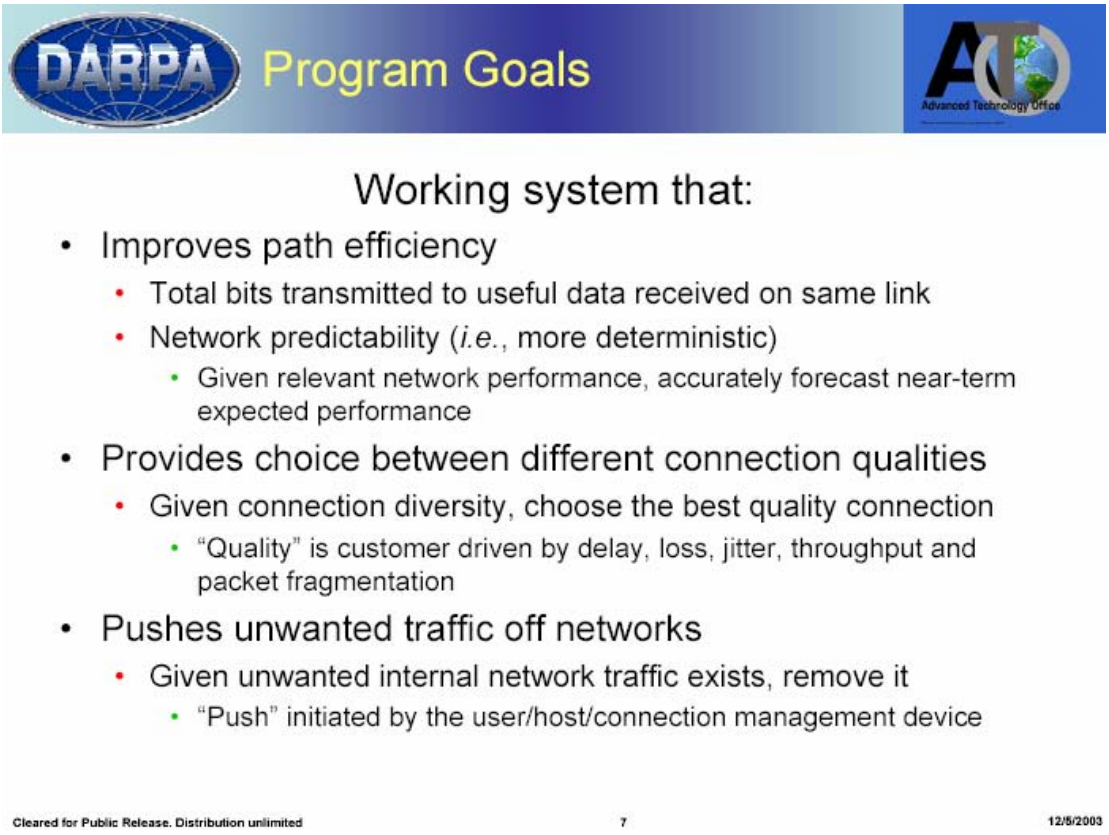
R^3 therefore provides a powerful new approach to robust networking that is able to address the increasing demands being placed on the network, whilst retaining the flexibility to cope with an unpredictable and rapidly changing infrastructure.

B.3 The challenge as described by DARPA

Some of DARPA's requirements for future networking technologies have been captured in a presentation[13], cleared for public release, by Col T Gibson. In particular, Gibson identifies program goals to improve path efficiency, provide choice between different

[13] 5th Dec 2003, Control Plane, Col T. Gibson:
http://www.darpa.mil/ato/solicit/ControlPlane/gibson_brief.pdf

connection qualities and to enable unwanted traffic to be pushed off the network, as shown in the figure below.



Path efficiency & Network Predictability

Path inefficiency is created by high levels of control information in order to ensure the delivery of the useful information. Dynamic routing, for example, links path recovery with re-convergence of the whole network. A rapid restoration time therefore requires more frequent flooding and higher levels of control information (and longer periods of network instability). MPLS, for example, requires connections to be set up and torn down across the network and, therefore, higher levels of control information. R^3 separates recovery and traffic management from re-convergence and so can flood less often reducing the levels of control information. R^3 uses connections that are derived from the network topology and so does not need control information to set up and tear down individual end-to-end connections.

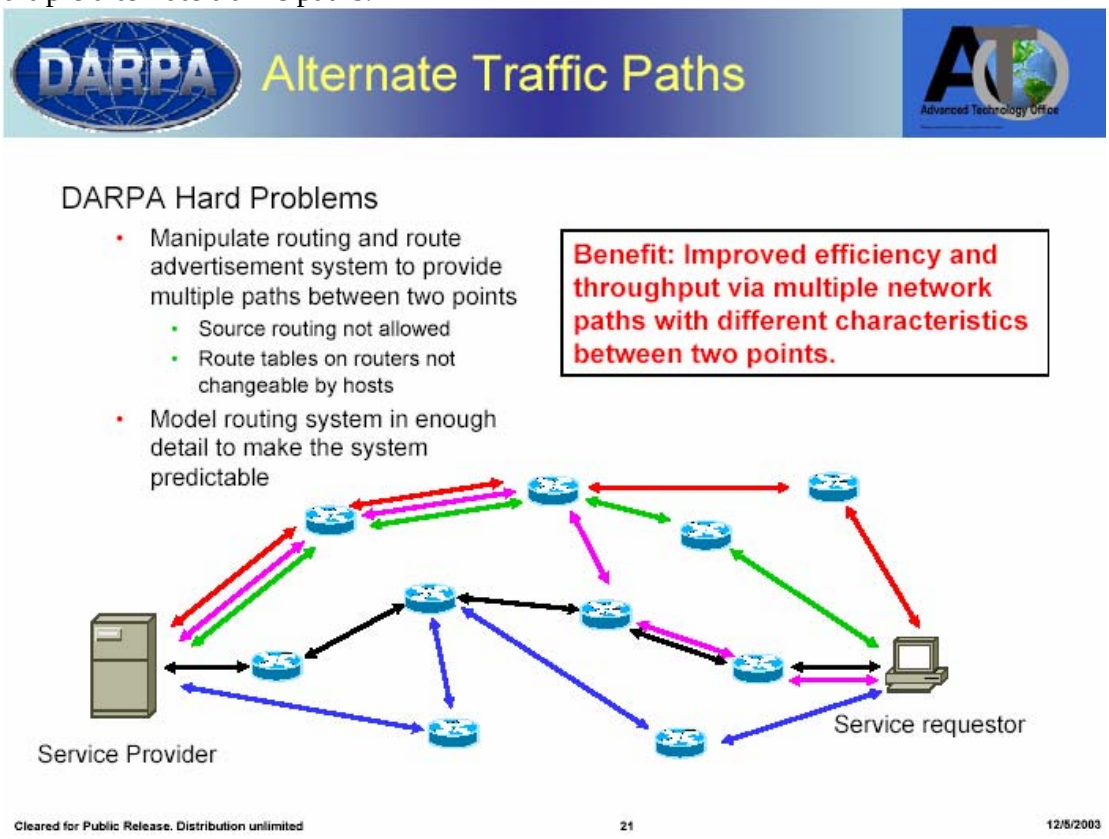
Current approaches to improving network predictability are required to manage the network using a single dimension tied to the physical topology. R^3's abstraction of the network allows it instead to be managed at multiple levels, with each level of increased abstraction managing the network over an increased time interval. Fast responses to local

traffic peaks can therefore be managed alongside a more progressive response to a more gradual change in traffic patterns over the network.

Connection Qualities & Low Priority Traffic

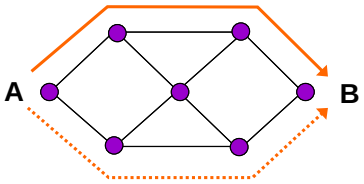
With Dynamic Routing, it is possible to expedite high priority traffic along a given path by placing it in a separate queue and then expediting its transmission out over the link. This approach though causes high levels of packet loss amongst the traffic in the lower priority queues, the delivery of which might not be so time critical, but may well be important. If we are not careful, too much of the traffic ends up being marked as a high priority and there is no scope for prioritisation, or too little traffic is marked as high priority and too much of the traffic suffers from packet loss.

R^3 offers an alternative way to address this issue, as there are usually multiple paths for each route through the network that can be exploited. It is, therefore, possible to map the different types of traffic onto the different paths through the network. For example, time critical traffic can follow paths with low latency, high importance traffic can follow paths that are less heavily loaded, and low importance traffic can be directed towards the remaining paths. This approach was identified by DARPA as one of the "hard problems" to solve – see figure below. We believe that R^3 provides a viable solution for achieving multiple alternate traffic paths.

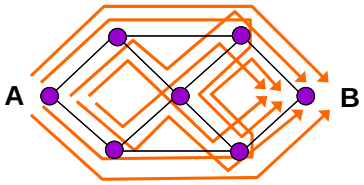


B.4 An overview of R^3

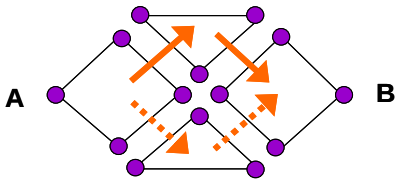
Earlier we saw that traditional networking technologies may provide two alternative paths across the network – for example, on the route between A and B in the figure below. In the case of Dynamic Routing, ECMP may provide two alternative shortest paths of the same cost. In the case of MPLS, quite often there will be both a primary and a secondary path created across the network.



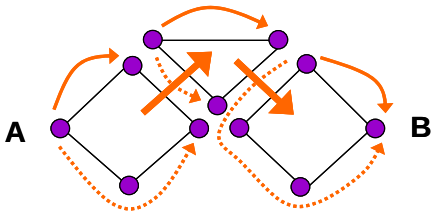
However, for the equivalent route in a network using R^3, there would be 8 alternative paths – as shown in the figure below.



This difference is created because the routes used in R^3 are based on a recursive abstraction of the network topology. Put in more practical terms, R^3 recognises that, as the data travels from A to B that it must pass from ring to ring to ring, either across the top or the bottom of the network – see the figure below.

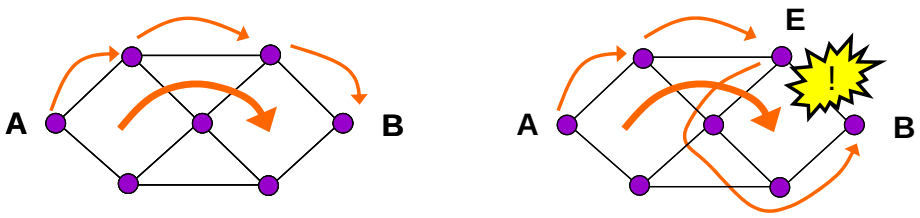


Furthermore, as the data travels across each ring, R^3 recognizes also that the data can take a path across the top or the bottom of that ring – see the figure below.



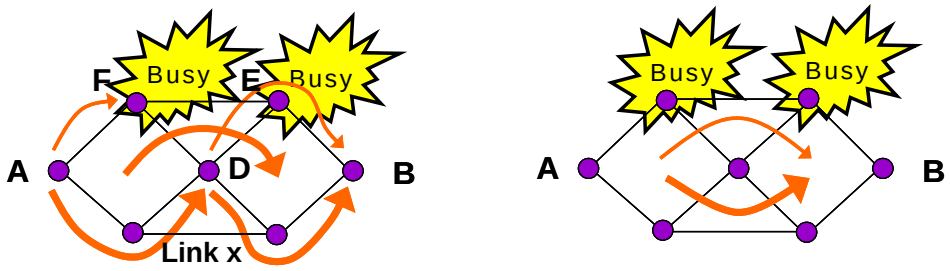
It should be apparent that the first choice of going round either the top or the bottom of the "ring of rings" is in fact a very similar to the second choice of going round either the top or bottom of the physical ring. The elegance of R^3 is that its recursive abstraction of the network makes the mechanism dictating each of these choices identical, and allows it to keep identifying and exploiting the inherent diversity in the network until the abstraction becomes loop free (i.e. there are no choices left) and routing is deterministic – hence its name, R^3: Resilient Recursive Routing.

The benefit of this approach can be seen by looking at how R^3 deals with a number of different scenarios. Consider a scenario, as shown in the figure below, where data is being routed from A to B by going clockwise around the ring of rings across the network, and by going clockwise around each individual ring. The net result is a path that goes from A to B across the top perimeter of the network.



Now let us assume that the link between E and B is lost. Data can still be routed from A to B by going clockwise around the ring of rings across the network, and by going clockwise around the first and second ring. In fact, the only thing that needs to change is that in order to reach B the data needs to be sent anti-clockwise around the final third ring – see the figure above. There has been no need to re-converge the network in order to find an alternative path – as with Dynamic Routing, and there has been no need to have a dedicated secondary path as with MPLS. A key strength of the R^3 architecture is therefore its ability to keep routing data in the face of significant attrition, without any need for re-convergence.

The power of the architecture is not just that it can exploit its lower level rings in order to recover quickly from link failures. The higher layers of abstraction allow the network to also adapt to changing network conditions and traffic patterns over progressively longer time intervals. For example, let us assume that both nodes E and F become congested and then stay congested over a longer period of time. In the short term, both A and D, detecting that the clockwise path around their rings is congested, will instead send the data anti-clockwise around their ring – see the figure below.

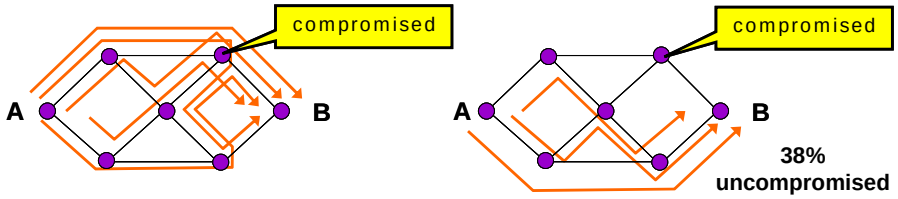


This is an excellent short-term solution to the congestion, but if it persists it makes sense to send a greater proportion of the traffic anti-clockwise around the ring of rings to make more use of the resources on the other side of the network such as link "x". It is at this next level of abstraction, over a longer time interval, that the persistent nature of the congestion is detected. The routing policy at this level is then adjusted and a greater proportion of the traffic is now sent anti-clockwise around the ring of rings.

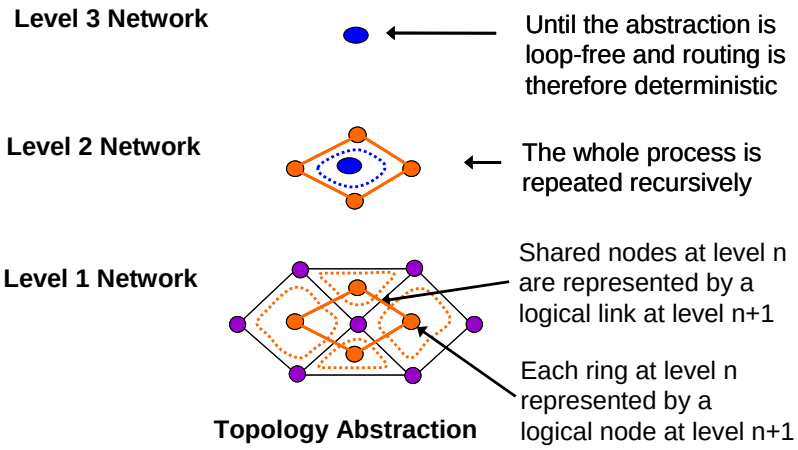
The strength of R^3 therefore is not only in its ability to respond quickly to deal with local occurrences of failure or congestion, but to also use its layers of abstraction to provide a progressive response to the state of the network over increasingly longer time intervals.

The creation of multiple alternative paths for each route in the network provides the opportunity for many other benefits in addition to the effective handling of network attrition or congestion. For example, particular links in the network could be reserved or prioritised for the use of particular traffic types. Other traffic types could be denied access to paths that make use of these key links, or could be bounced off them onto other paths when traffic of higher precedence requires them– see the figure below.

Another example, of how the multiple alternative paths could be used is in enhancing security. The packets that make up a secure transmission could be randomly routed over all of the possible paths. Then if a particular node was compromised, for example node E would only receive 62% of the message packets, as the other 38% of the transmission would be carried on paths that did not go through the compromised node.



R^3 provides a unique approach to networking because it performs routing based on a recursive abstraction of the network topology. The figure below shows the recursive abstraction of the simple network that has been used throughout this overview. The level 1 network represents the actual physical network. Each of the rings at level 1 is represented by a logical node at level 2. Nodes that are shared by two rings, act as gateways between those rings and are represented by links between the nodes at level 2. Once the level 2 network is abstracted from the level 1 network, the whole process is repeated recursively until a level is reached that possesses a completely loop-free structure and routing is now deterministic.



Appendix C: A Specific R^3 Routing Implementation

Even though we can devise a topology discovery and destination advertisement mechanism based on R^3, we choose to adopt for simplicity a standard link-state routing protocol such as IS-IS or OSPF, to achieve both of these network functions. This is done in order to concentrate on routing alone. The numbering/naming of higher-level nodes can be implemented in a variety of ways, but as the details are not central to this paper, they are omitted for brevity.

Routing is achieved by employing labels hereafter called *circulation vectors*, which are also implemented recursively (i.e. they are nested in the header of each packet). For a level *k* destination each circulation vector describes a *local* level 1 simple path which is a subgraph of the local level 1 cycle (i.e. loop segment or 'arc' on the local level 1 loop) towards the destination, a level 2 'arc' on the local level 2 loop towards the destination, etc., all the way up to a 'local' level *k* 'arc' on the 'local' level *k* loop, containing the destination. Note that this routing scheme is not the same as source routing[14], as it does not specify a route to the destination, but rather a progressively abstracted route to the destination. This provides a connectionless service that gives specific physical path selection on the shortest time-scale level 1 neighborhood, but as a result of the increasing levels of abstraction provides more flexibility in subsequent physical path selection across any remaining longer time-scale higher-level neighborhoods. This retained flexibility is then used at subsequent nodes to make local forwarding decisions in order to overcome any congestion and failure situations that arise.

A selected loop segment or hop at level *n* requires that the packet be forwarded from one node to an adjacent node using a link, all at level *n*. Each node at level *n* is in fact a representation of a neighborhood at level *n*-1 within a planar topology. Therefore, the link at level *n* is in fact a representation of the nodes held in common between two adjacent neighborhoods at level *n*-1. These common nodes are gateways and thus represent an intermediate destination at level *n*-1 of a selected path at level *n*. Nodes receiving a packet will forward the packet so as to maintain its given direction of circulation on the designated ring, until it reaches the gateway. Once the packet reaches a gateway, the circulation vectors of all completed hops are removed and new ones are added, based on more recent information regarding congestion and even failures, until the packet is routed to its final destination.

The above procedure can be illustrated using the simple 3-level network of Fig. C.1. In sending a packet from a host A connected directly to node 0.1 to a host B connected directly to node 0.11, host A generates a packet with destination address B. Node 0.1 will have knowledge of the existence of B through the advertisement protocol only as a level

[14] Tanenbaum, A. S., Computer Networks, PRENTICE-HALL, London, Chapter 5, p.415-416

3 destination attached to the level 3 node 2.3. As the level 3 network description is a simple tree, the routing on it is deterministic and we omit the use of level 3 circulation vectors in our discussion for simplicity.

The routing required at level 3 is from neighborhood 2.1, which node 0.1 belongs to, to neighborhood 2.3, which node 0.11 belongs to, with the next hop being neighborhood 2.2. The link from 2.1 to 2.2 is represented by the nodes that 2.1 and 2.2 have in common at the next lower level, i.e. nodes 1.2 and 1.3. At level 2 there is, therefore, path diversity, as node 0.l may send the packet, either clockwise around ring 2.1 to gateway 1.2, or anticlockwise around ring 2.1 to gateway 1.3. Node 0.1 selects one of these two paths, for example ring 2.1 clockwise to gateway 1.2, based on *summarized performance* information from around the level 2 ring 2.1 *on a longer time-scale*, and attaches an inner label containing the selected circulation vector to the packet.

The routing required at level 2 is to forward the packet from neighborhood 1.1 to neighborhood 1.2. The link from ring 1.1 to ring 1.2 is represented by the nodes that 1.1 and 1.2 have in common as the next lower level, i.e. nodes 0.2 and 0.3. At level 1 there is, therefore, path diversity, as node 0.1 may send the packet, either clockwise around ring 1.1 to gateway 0.2, or anticlockwise around ring 1.1 to gateway 0.3. Node 0.1 selects one of these two paths, for example ring 1.1 clockwise to gateway 0.2, based on *measured performance* information from around the level 1 ring 1.1 *on a shorter time-scale*, and attaches an outer label containing the selected circulation vector to the packet.

The routing required at level 1 is now to forward the packet from node 0.1 to node 0.2. As the link from 0.1 to 0.2 corresponds to a physical link between these nodes, there is no further path diversity that can be exploited and the packet is forwarded along the physical link to node 0.2.

Denoting positive circulation around a ring to be clockwise (this does not need to be unique other than for the member nodes of that ring), a possible packet structure corresponding to the first routing decision, shown in Fig. C.2 , would be Label 2 (inner label): 2.1+ to 1.2 and Label 1 (outer label): 1.1+ to 0.2. When this packet arrives at the level 1 node 0.2, this node identifies itself as being 0.2 the destination gateway of the outer label and so strips the outer label. It also identifies itself as being a member of the neighborhood 1.2 the destination gateway of the inner label and so strips the inner label as well.

This occurs because neither of the circulation vectors is required in addition to the destination host address B to ensure deterministic routing. Indeed, it is quite acceptable to adopt a policy of penultimate node label stripping, so that labels are stripped if the adjacent node that the packet is being sent to is in fact the label destination. Labels are, therefore, only needed in order to ensure packets are correctly transited through intermediate nodes at all levels in the abstraction. There was therefore no need to add any labels to the packet leaving node 0.1. However, for clarity, all labels will continue to be

shown throughout this example.

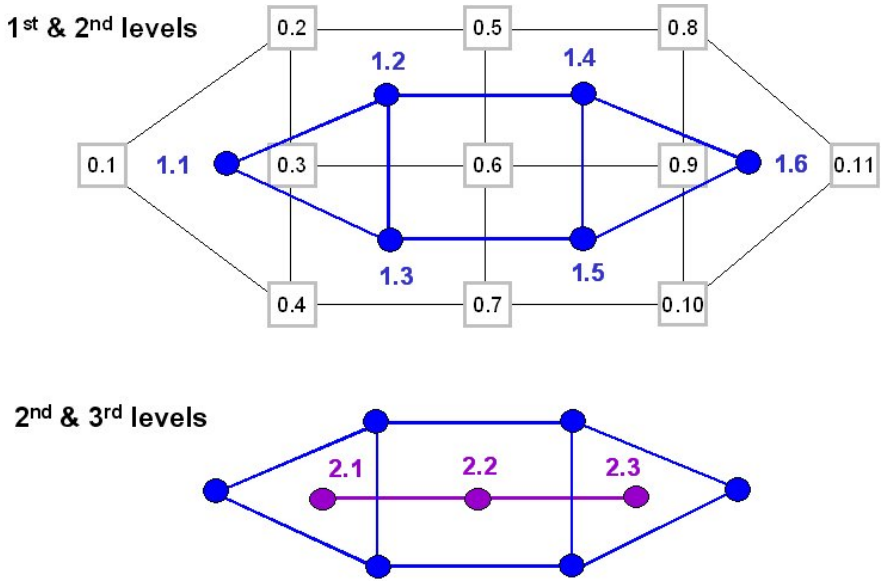


Fig. C.1 *Routing on a simple network*

Node 0.2 follows the same process of establishing the associated path diversity and then making path selections based on performance information associated with each level in the abstraction. The packet leaves node 0.2 towards 0.3, for example, with an inner label, abbreviated as L2, of 2.2- to 1.5 and an outer label, L1, of 1.2- to 0.3.

Upon reaching node 0.3, the outer L1 label has reached its destination and is removed, but the inner L2 label has not and so it is retained and the next level 1 path is selected. Node 0.3 must maintain the circulation at level 2 of 2.2- to 1.5 and as node 0.3 belongs to neighborhood 1.3, it must forward the packet along 2.2- from gateway 1.3 to gateway 1.5. As the common nodes between rings 1.3 and 1.5 are gateways 0.6 and 0.7, node 0.3 can forward the packet either on 1.3+ to 0.6 or 1.3- to 0.7. Node 0.3 selects one of these two paths, for example1.3+ to 0.6, based on the most recent level 1 performance (e.g. congestion) information.

Upon reaching node 0.6, both the outer and inner labels have reached their destination and are thus removed. New labels are inserted following the same process that occurred at node 0.2.

Upon reaching node 0.7, neither the outer label L1 nor the inner label L2 destinations have been reached and 0.7 simply maintains both circulation vectors and the packet is forwarded without choice to node 0.10 without any change to the labels.

Upon reaching node 0.10, the outer label L1 and the inner label L2 have both reached

their destination and are removed. Node 0.10 has knowledge of the existence of B through the advertisement protocol as a level 1 destination as nodes 0.10 and 0.11 are both members of neighborhood 1.6. Node 0.10, therefore, follows the same process but only has to consider whether to send the packet on 1.6+ to 0.11 or 1.6- to 0.11. In this example, the packet is forwarded on 1.6- to 0.11 based on the most recent level 1 performance information only.

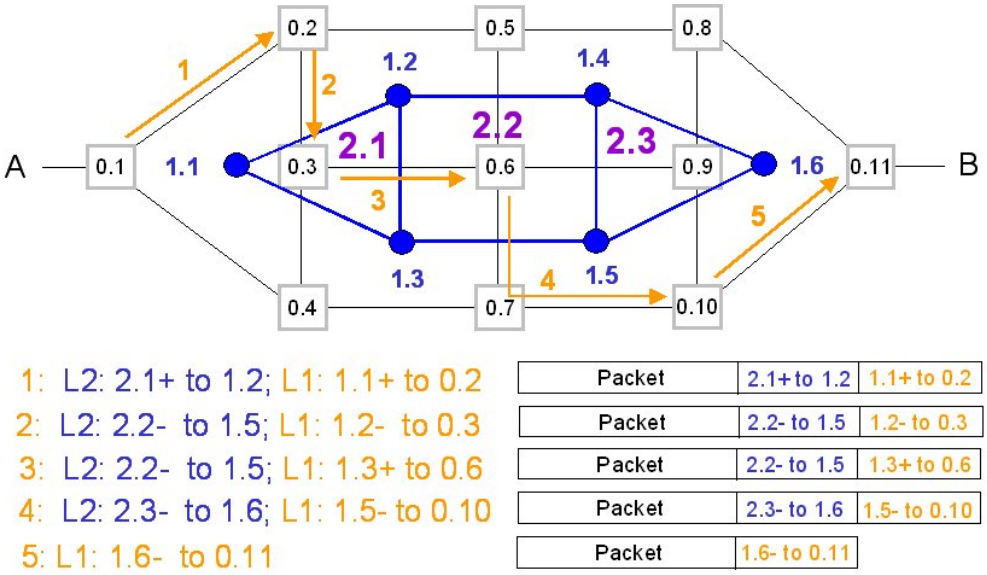


Fig. C.2 *A set of routing decisions on the network of Fig. C.1*

The simplest performance information we employ in our simulations (see §3) is the measured cumulative delay a modified "hello" packet experiences per hop in traversing each loop in each of the circulation directions approximately every 100 ms. In our simulations we currently use static hop counts as the higher-level "summarized performance" information. Future work will implement the level 2 summarized performance to be the average loop traversal delay over both circulations for 100 level 1 measurements, disseminated through restricted flooding every 10 seconds.

We want to stress here that labels are not path specific and that performance information at level 2 is summarized and disseminated over a longer time-scale (not specified, nor implemented here) than performance information at level 1, thus making the routing protocol highly dynamic in its switching decisions. Moreover, we can afford to cleanly disassociate the time scales that underlie the topology discovery, path choice and forwarding decision mechanisms.

Appendix D: R^3 OPNET modules

All our simulations were implemented using the industry-standard network modeling environment, OPNET® Modeler.

Fig. D.1.A shows the OPNET R^3v8 node module. Each node has a traffic generator (associated with the identical user traffic generation rate), an input queue, an R^3v8 processor and a number of output queue and transceiver interfaces, corresponding to the upper bound of the node degree.

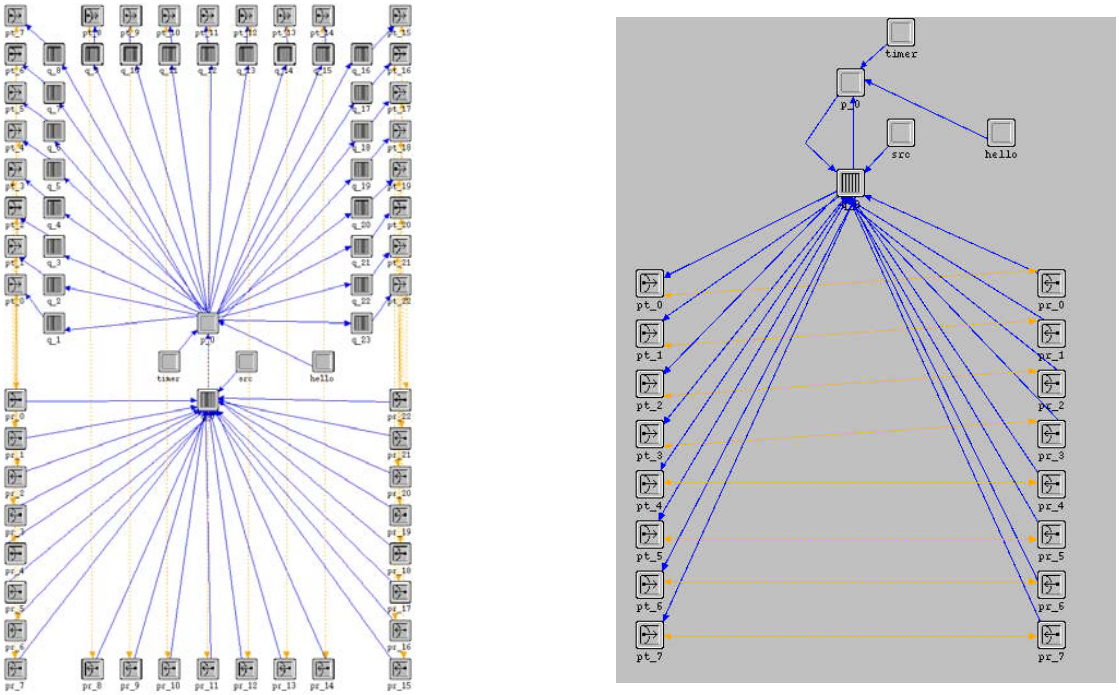
Fig. D.1.B shows the OPNET R^3v9 node module. Because the output queue throughput at one end node of a point to point connection directly affects the performance of the input queue at the other end node of the connection. By appointing a single queue for both input and output traffic at each node, we try to simulate the scenario where the throughput of a node is heavily affected by that of other nodes in its neighbourhood. As we know that in a wireless network, every node has to compete against each other to access a common medium. Due to time and resources limited, we have not simulated a multiple access control mechanism in MAC layer, such as CDMA (Code Division Multiple Access) or TDMA (Time Division Multiple Access).

The queue, as shown in Fig. D.2, is processed according to a basic First In First Out (FIFO) policy. Each queue has multiple sub-queues rather than only one, because when there is only one sub-queue, the routing state traffic (i.e. signalling traffic) can then be delayed by a heavy user traffic load in the simulation. The queue processor services packets from various sub-queues with different frequencies. Obviously, the higher the service frequency for a sub-queue, the higher the priority for the corresponding type of traffic is. Then, since signalling traffic (sub-queue 0) has the highest priority, the user traffic may be delayed in being serviced when the routing state traffic is heavily loaded, but not the other way around. In order to avoid heavy signalling traffic, constraint flooding is used at current R^3v8 logical level routing state broadcasting.

Input-queues are modelled as having effectively an infinite capacity, so that the routing box itself is treated as non-blocking. Discarded packets arise solely from link congestion (output-queue overflow) or routing errors, such as a routing loop or inaccessible next hop. When a routing loop occurs, packets forwarded more than the maximum permissible hop count (i.e. age) are dropped. When the information about some link or node failures is not disseminated properly, a packet that cannot be forwarded to the destination will be discarded. The discarded packets are recorded to calculate the total number of lost packets of the simulated algorithm.

The processor module is shown in Fig. D.3. Each processor has a unique identity and several process states.

- In the "init" state, the routing processors load the initial R^3 architecture, calculate R^3 routes, and then generate their own R^3 routing tables automatically.
- In the “idle” state, the processors keep quite and wait for incoming packets or interrupts.
- Upon a node/link failure/recovery interrupt, the corresponding part of the routing table will be updated at the "NB-FR" state. The consequent new routing states could be generated and constraint flooded to a set of nodes in the relative rings or neighbourhood.
- Upon a packet or routing states arrived, the data units will be processed in the "RCV" state. The flowchart of the “RCV” state is shown in Fig. D.4.



Multiple queue node module Single queue node module

Fig. D.1: *OPNET node module*

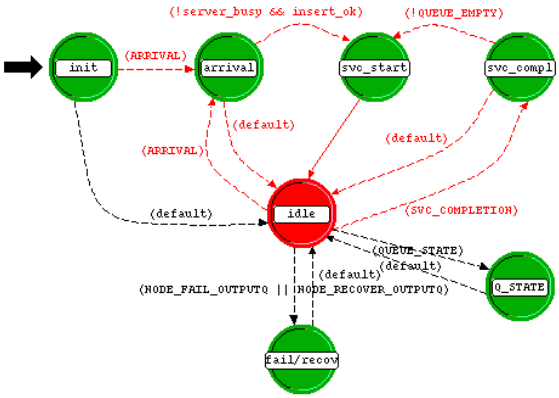


Fig. D.2: *OPNET R[3]v8 queue module*

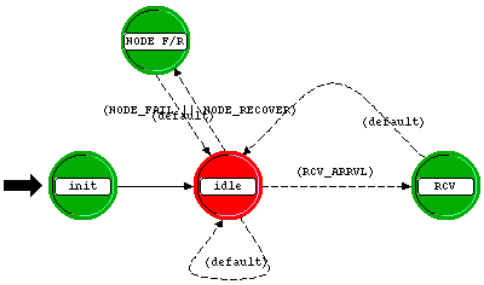


Fig. D.3: *OPNET R[3]v8 processor module*

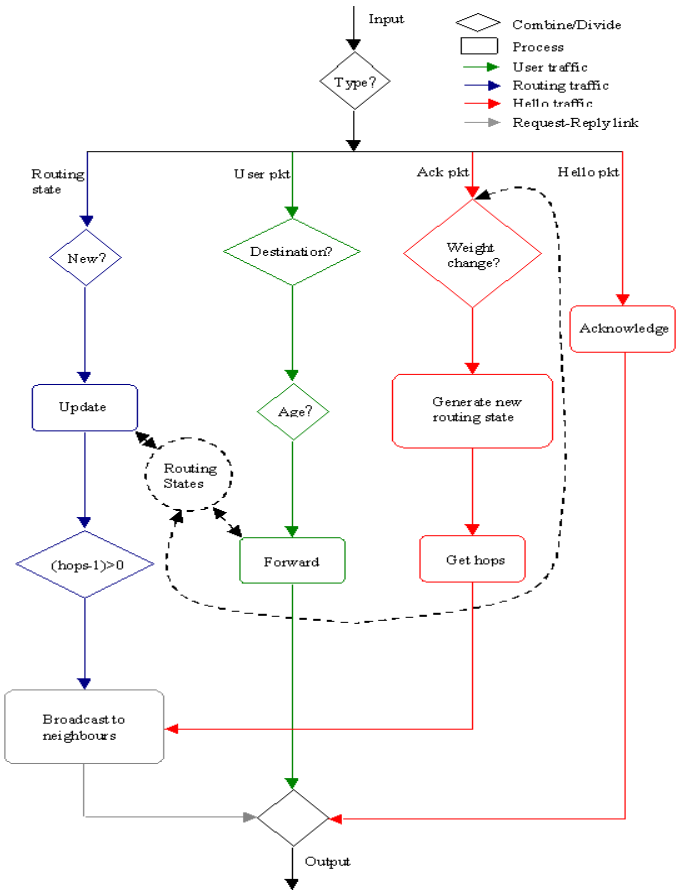


Fig. D.4: *Flowchart of "RCV" state*

Appendix E: R^3 OPNET Simulation Settings

Simulation I

Each level 1 link has a 2.5 Gb/s full-duplex capacity. Each router is non-blocking and has queue buffers corresponding to 200 ms of traffic at the notional link speed. All nodes generate packets of fixed 512 byte size, 2/3 of which are randomly addressed to the even numbered nodes in the network and 1/3 of which are addressed to node 15 (the hot spot).

The network was simulated for 6 s, with a level 1loop performance measurement period of 90 ms. We found that it was not possible to simulate the network in OPNET® in any reasonable time-period due to the fact that simulations took many days to run on a SUN® Sparc-station Ultra 10 with 2Gbytes of RAM. For this reason, we implemented bit and time scaling, where 1 simulation bit corresponds to 25 bits in reality and 1 simulation second corresponds to 0.03 seconds in reality. Using these scaling laws, links are scaled to 3 Mbps and buffer queues can accommodate 2000 packets before overflowing. We choose to model buffers as infinite in the simulation, but consider the 2000 packet threshold in discussing the packet loss rate. It is not possible to scale the 512-byte packets to 164 bit packets because of the amount of information we wish to convey in our simulation. Therefore, as a compromise we set the packet size to be 10,000 bits after scaling, which is equivalent to routing a burst of packets.

Simulation II

We start with three fully connected nodes and add a further 37 nodes according to the Albert-Barabasi algorithm, each of which has either one or two links with equal probability. The resulting network has 12 stubs and is planar. Each level 1 link has a 10 Mb/s full-duplex capacity. Each router is non-blocking and has queue buffers corresponding to 400 ms of traffic at the notional link speed. The data packet size is fixed at 512 bytes. The network was simulated for 80 s, with a level 1 loop performance measurement period of 400 ms. The node failure took place at node 0.3 at a simulation time of 20 s. Node 0.3 is the second most highly connected node in the network and as such is a good candidate for an asymmetric attack. Node 0.5 is the most highly connected node, but decimating this results in a network that is partitioned and no routing algorithm can possibly cope with such an event.

Bit and time scaling is implemented, where 1 simulation bit corresponds to 4 bits in reality and 1 simulation second corresponds to 0.4 seconds in reality. Links are scaled to 1 Mbps and buffer queues can accommodate 1000 packets before overflowing. The whole simulation lasts 200 simulation seconds and node 0.3 fails at the 50th simulation second.

Simulation III

Each level 1 link has a 10 Mb/s full-duplex capacity. Each router is non-blocking and has queue buffers corresponding to 800 ms of traffic at the notional link speed. The user packet size is fixed at 512 bytes. The network was simulated for 10 s, with a level 1 loop performance measurement period of 100 ms. The adaptation of R^3v8 at level 2 was based on a summarized level 1 performance information every 500 ms. The node failures took place at node 0.2 and node 0.30 at the 3rd and the 6th second.

Only time scaling is implemented (as the simulation modules have been improved and simulation III became more manageable with our available simulation resources), where 1 simulation second corresponds to 0.1 seconds in reality. Links are scaled to 1 Mbps and buffer queues can accommodate 2000 packets before overflowing. The whole simulation lasts 100 simulation seconds and nodes 0.2 and 0.30 fail at the 30th and the 60th simulation second respectively.

Simulation IV

For ISIS and R^3 networks, where single queue node module was introduced to simulate wireless connection, each physical link has a 1 Mb/s full-duplex capacity. The AODV network was simulated by standard OPNET wireless node module with a CDMA MAC layer, where packets can be collided and forced to be retransmitted. Each node has a "direct sequence" wireless interface with a capacity of 5.5 Mbps.

Other settings for the standard OPNET AODV node module include:

- Node Traverse Time 1*E-5 s
- AODV Hello Interval uniform (0.1, 0.001) with Allow Hello Lost 30 times
- WLAN Power 0.001W
- Attributes of wlan_port_rx_0_0: noise figure 1.0, ecc threshold 1.0, and null ragain model, bkgnoise model, inoise model, ber model, or error model
- Wireless LAN short/long retry limit 7/4, max receive time 0.5sec., buffer 100Mbits

The convergence time for both R^3 and ISIS is identically 3 seconds. Due to the improvement of simulation conditions, time scaling or bit scaling was not introduced in this simulation work.